The Embedding Method

or

How to Handle Locally Broken Symmetries

Jeroen van Hoof

The Embedding Method or How to Handle Locally Broken Symmetries

The Embedding Method
or
How to Handle Locally Broken Symmetries

een wetenschappelijke proeve op het gebied van
de Natuurwetenschappen

Proefschrift

ter verkrijging van de graad van doctor
aan de Katholieke Universiteit Nijmegen
volgens besluit van het College van Decanen
in het openbaar te verdedigen op
maandag 3 februari 1997,
des namiddags om 3.30 uur precies
door

Jeroen Bastiaan Antonius Nicolaas van Hoof

geboren op 30 juli 1969
te Oudenbosch

Promotor: Prof. Dr. J.E. Inglesfield

Manuscriptcommissie: Prof. Dr. R.A. de Groot
Dr. S. Crampin (University of Bath)
Dr. P.J. Kelly (Philips Research, Eindhoven)

ISBN: 90-9010174-8

Het werk beschreven in dit proefschrift maakt deel uit van het onderzoeksprogramma van de Stichting voor Fundamenteel Onderzoek der Materie (FOM) en is mede mogelijk gemaakt door financiële steun van de Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO).

Voorwoord

Dit is de plaats in het proefschrift die traditioneel gebruikt wordt om iets te zeggen over de niet-wetenschappelijke kanten van het werken aan een proefschrift. En hoewel op een proefschrift vaak slecht één naam staat, zijn er natuurlijk altijd anderen bij betrokken.

Als eerste wil ik mijn promotor John Inglesfield bedanken, die mij gedurende mijn promotie-onderzoek (en ook al tijdens mijn afstudeerwerk) zeer goed en enthousiast heeft begeleid. Zijn enthousiasme was soms ook hard nodig, als er voor de zoveelste maal weer een fout zat "ergens" in de matrix-elementen.

Verder wil ik Simon Crampin en Maziar Nekovee bedanken voor de samenwerking gedurende de eerste jaren van mijn promotie-onderzoek, waarin we met ons vieren nieuwe manieren hebben ontwikkeld om tegen 'embedding' aan te kijken. Simon wil ik ook bedanken voor het zeer kritisch doorlezen van mijn manuscript.

Ook wil ik Kees Schep en Paul Kelly bedanken voor de samenwerking die ontstaan is naar aanleiding van de vraag of ik de verstrooiings-matrices, die ik ergens diep in de computer-code uitreken, niet zou kunnen gebruiken om te rekenen aan geleidings-vraagstukken. Deze samenwerking heeft geleid tot de geleidbaarheids-toepassingen zoals beschreven in hoofdstukken 4, 9, 12, 13 en 14. Deze toepassing zijn vrij nieuw en ik hoop hieraan in de toekomst verder te kunnen werken. Ook wil ik Paul bedanken voor het doorlezen van mijn manuscript.

Verder wil ik Rob de Groot bedanken voor het doorlezen van mijn manuscript, maar vooral voor de kritische opmerkingen over door mij gedane uitspraken aan de koffietafel. Dit heeft er vaak toe geleid dat ik iets beter over bepaalde dingen heb nagedacht.

Verder wil ik alle andere mensen bedanken die tijdens mijn promotie-onderzoek met mij in aanraking zijn gekomen, waaronder natuurlijk alle mensen van Theoretische Fysica, waarvan ik met name Bea Honer en Rina Vos wil noemen.

Naast de tot nu toe genoemde personen, waarmee ik voornamelijk (doch niet uitsluitend) tijdens mijn werk in aanraking kwam, zijn daar natuurlijk ook mijn vrienden, Edwin Boon, Hervé Kuijten en Rik Fleuren die voor de na hard werken zo noodzakelijke ontspanning hebben gezorgd.

Aan mijn ouders, die voor een belangrijk gedeelte verantwoordelijk zijn voor de vorming van mijn karakter en persoon, wil ik hierbij mijn dank betuigen. Zij hebben mij altijd gestimuleerd om te proberen net iets verder te gaan dan mijn kunnen. In deze dank wil ik ook mijn broer Paul betrekken.

Tenslotte wil ik de persoon bedanken, die ik via mijn werk heb leren kennen (waar internationale wetenschappelijke conferenties al niet goed voor zijn), maar die nu het belangrijkste deel van mijn leven uitmaakt samen met ons zoontje Tycho. Martina .... bedankt voor alle goede dingen in mijn leven in de afgelopen jaren *en* hopelijk nog vele jaren die komen gaan.

Nijmegen, december 1996 Jeroen van Hoof

As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality.

Albert Einstein

Most people see things that are and say, "Why?" But I dream things that never were and I say, "Why not?"

George Bernard Shaw, "Back to Methuselah"

The reasonable man adapts himself to the world; the unreasonable one persists in trying to adapt the world to himself. Therefore all progress depends on the unreasonable man.

George Bernard Shaw

For every problem, there is one solution which is simple, neat and wrong.

H. L. Mencken

The most exciting phrase to hear in science, the one that heralds new discoveries, is not "Eureka!" (I've found it!), but "That's funny..."

Isaac Asimov

Time sneaks up on you like a windshield on a bug.

Jon Lithgow

There is a theory that states: "If anyone finds out what the universe is for it will disappear and be replaced by something more bizzarly inexplicable." There is another theory that states: "This has already happened..."

Douglas Adams, "Hitchhiker's Guide to the Galaxy"

As far as I'm concerned, if something is so complicated that you can't explain it in 10 seconds, then it's probably not worth knowing anyway.

Calvin, "Calvin and Hobbes" by Bill Watterson

aan Martina, Tycho en mijn ouders

Contents

Chapter 1

Introduction

The aim of this first chapter, is to introduce the field of electronic structure calculations and outline its place within solid-state physics. I will list some of the standard ingredients of most methods for doing such calculations. When it comes to handling non symmetric systems there are some frequently used "tricks", which have both their advantages and disadvantages. Next I will shortly say something about the advantages and disadvantages of the trick that I use to handle locally broken symmetries.

1.1 Electronic Structure Calculations

In the field of solid state physics there has always been an interest in the microscopic structure of materials. By microscopic I mean on the level of the atoms. Since it is impossible to measure this structure directly in experiment, one always has to perform some analysis of what can be measured. Even those experimental methods that seem to measure (atomic) structure directly (STM) will frequently (usually) need some theoretical interpretation. To perform this analysis or interpretation, one has to have an idea how the microscopic system will behave. This is one of the reasons for using the theory of quantum mechanics to describe materials microscopically.

On the microscopic scale the electrons' behavior is what shapes the material, which is why most of the calculations focus on what makes the electrons tick.

Nowadays, electronic structure calculations are used within solid state physics not only to help with the analysis of experiments, but also to predict interesting new materials with certain desired properties. As such these calculations have become powerful tools, which will grow in usefulness as computers are getting more efficient and methods more elaborate.

Why the power of computers has put such a limit on the calculations will become clear in the following section.

1.2 Some standard ingredients

In this section a short description will be given of some of the ingredients found in most methods for doing electronic structure calculations.

1.2.1 Schrödinger equation

The basic ingredient of most electronic structure calculations is the very simple and elegant equation:

$$H\Psi = E\Psi \tag{1.1}$$

This equation is usually the Schrödinger equation known from quantum mechanics. The fun starts when we want to define what H, E and Ψ are.

Ψ is the wavefunction, and in the case of a solid, we are talking about the wavefunction of a system of the order of 10^{23} interacting particles. H is the Hamiltonian of the system. Again this is the Hamiltonian of a system of 10^{23} interacting particles. E is the energy of the system in the particular state associated with wavefunction Ψ. All this means that both Ψ and H in (1.1) are functions of the coordinates of 10^{23} electrons *plus* 10^{23} nuclei.

One of the first steps in most derivations of the theories behind the methods for performing calculations is to limit the particles involved to just the electrons, assuming that the atomic nuclei do not respond as quickly as the much lighter electrons. This is the first approximation I will discuss briefly.

1.2.2 Born-Oppenheimer approximation

When we write out the Hamiltonian of (1.1) in a real space notation, with both the atomic nuclei and the electrons as interacting particles, we get the following[1]:

$$H = -\sum_l \frac{1}{2M_l}\nabla_l^2 - \sum_i \frac{1}{2}\nabla_i^2 + \frac{1}{2}\sum_{i\neq j}\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + U(\mathbf{u},\mathbf{r}) + V(\mathbf{u}) \tag{1.2}$$

where $\mathbf{u}$ denotes all nuclei coordinates and $\mathbf{r}$ all electron coordinates. The first term is the kinetic energy of the nuclei (mass M_l) and the second term the kinetic energy of the electrons. The third term is the electron-electron interaction, and therefore excludes the $i = j$ term (i.e. no self-interaction). The $U(\mathbf{u},\mathbf{r})$ term is the electrostatic interaction between electrons and nuclei. The term $V(\mathbf{u})$ contains all the electrostatic interactions between nuclei.

We use the *ansatz* that we can decouple the electron wavefunction from the nuclear-wavefunction, i.e.

$$\Psi(\mathbf{u},\mathbf{r}) = \chi(\mathbf{u})\Psi_{\mathbf{u}}(\mathbf{r}) \tag{1.3}$$

This *ansatz* is normally valid because the nuclei are so much heavier than the electrons, and therefore the electron wavefunction can easily adjust to nuclear motion. Thus we can put the motion of the nuclei in $\chi(\mathbf{u})$ and put the electron wavefunction in $\Psi_{\mathbf{u}}(\mathbf{r})$, which is the electron wavefunction evaluated with the nuclei at the positions given by $\mathbf{u}$. It follows that $\Psi_{\mathbf{u}}(\mathbf{r})$ satisfies:

$$\left[\sum_i -\frac{1}{2}\nabla_i^2 + \frac{1}{2}\sum_{i\neq j}\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + U(\mathbf{u},\mathbf{r})\right]\Psi_{\mathbf{u}}(\mathbf{r}) = \mathcal{E}(\mathbf{u})\Psi_{\mathbf{u}}(\mathbf{r}) \tag{1.4}$$

where the energy of the electrons is denoted by $\mathcal{E}(\mathbf{u})$.

[1] Throughout this thesis I will use atomic units, in which $e = \hbar = m_e = 1$. The atomic unit of length is the Bohr radius (0 5292 Å), and the a u of energy is the Hartree (27 211 eV)

We now put (1.3) into (1.1) using the Hamiltonian from (1.2), and using (1.4) we obtain:

$$\Psi_{\mathbf{u}}(\mathbf{r})\left[\sum_l -\frac{1}{2M_l}\nabla_l^2 + \mathcal{E}(\mathbf{u}) + V(\mathbf{u})\right]\chi(\mathbf{u}) - \chi(\mathbf{u})\sum_l \frac{1}{2M_l}\nabla_l^2\Psi_{\mathbf{u}}(\mathbf{r}) - \sum_l \frac{1}{M_l}\nabla_l\chi(\mathbf{u})\cdot\nabla_l\Psi_{\mathbf{u}}(\mathbf{r}) = E\chi(\mathbf{u})\Psi_{\mathbf{u}}(\mathbf{r}) \tag{1.5}$$

Now we assume:

$$\sum_l \frac{1}{2M_l}\nabla_l^2\Psi_{\mathbf{u}}(\mathbf{r}) = 0 \tag{1.6a}$$

$$\sum_l \frac{1}{M_l}\nabla_l\chi(\mathbf{u})\cdot\nabla_l\Psi_{\mathbf{u}}(\mathbf{r}) = 0 \tag{1.6b}$$

This is the Born-Oppenheimer or adiabatic approximation. This approximation leaves the nuclear wavefunction to satisfy:

$$\left[-\sum_l \frac{1}{2M_l}\nabla_l^2 + \mathcal{E}(\mathbf{u}) + V(\mathbf{u})\right]\chi(\mathbf{u}) = E\chi(\mathbf{u}) \tag{1.7}$$

We see that the kinetic plus potential energy of the electrons for a certain set of nuclear coordinates, $\mathcal{E}(\mathbf{u})$, acts as an extra potential energy term in (1.7).

The most important result of the Born-Oppenheimer approximation is that we can solve the electronic Schrödinger equation for a fixed configuration of nuclear positions. We have decoupled motions. Using the Born-Oppenheimer approximation we have gotten rid of a non-negligible fraction of our interacting particles. However even with only the electrons we still have to consider of the order of 10^{23} particles for a macroscopic piece of material. In the next section I will discuss a description which we can actually handle: this is Density-Functional Theory (DFT).

1.2.3 Density Functional Theory

Density-Functional Theory (DFT) is based on a theorem by Hohenberg and Kohn on the electron gas [1].

Take a number (N) of electrons, which interact via the Coulomb potential and feel an unspecified external potential. We treat the electrons as fermions, and neglect spin for the moment. Since the number of electrons is fixed, and we can only vary the external potential, $V_{ext}(\mathbf{r})$ defines the ground state of the system and therefore also the ground-state density $\rho_0(\mathbf{r})$ of the electrons.

Hohenberg and Kohn showed the reverse to be true. Given a ground-state density, then we know the external potential, which is unique for this ground-state density. This is Hohenberg and Kohn's first result:

> *The external potential of an electron gas is a unique functional of the electron density of the ground state (if the ground state is non-degenerate).*

From this it is clear that once we know the ground-state density, we know the complete Hamiltonian, energies and wavefunctions of all states.

Let us look at a density $\rho_0'(\mathbf{r})$ with external potential $V_{ext}(\mathbf{r})$ which is not the ground-state density Then, according to Hohenberg and Kohn's first result, this density corresponds to a new external potential $V_{ext}'(\mathbf{r})$, with $\rho_0'(\mathbf{r})$ as its ground-state density In other words, a density which is not a ground-state density of $V_{ext}(\mathbf{r})$ is always a ground-state density of some other external potential $V_{ext}'(\mathbf{r})$ We also have a ground-state wavefunction $\Psi'(\mathbf{r})$, belonging to $V_{ext}'(\mathbf{r})$ So given an external potential $V_{ext}(\mathbf{r})$, we define the energy-functional $E_V[\rho_0'(\mathbf{r})]$ as the energy belonging to the wavefunction $\Psi'(\mathbf{r})$

$$E_V[\rho_0'(\mathbf{r})] = \langle\Psi'|H_V|\Psi'\rangle \equiv \int \mathrm{d}^3\mathbf{r}\rho_0'(\mathbf{r})V_{ext}(\mathbf{r}) + F[\rho_0'(\mathbf{r})] \tag{1 8}$$

where we have put the kinetic energy and electron-electron interactions into the functional F We also have the boundary condition that the total number of electrons remains N, i e we always ensure

$$\int \mathrm{d}^3\mathbf{r}\rho(\mathbf{r}) = N \tag{1 9}$$

Upon study of (1 8) it becomes clear that the energy functional is minimal for the ground-state density $\rho_0(\mathbf{r})$, with the ground-state energy, and for all non-ground-state densities the energy functional will give a higher energy This can be summarized as

The energy of a fixed external potential can be written as a functional of the corresponding density

The density that minimizes this energy functional is the ground-state density

Let us examine the energy functional in a more explicit form To make progress we will consider the system of non-interacting electrons with the same density $\rho_0(\mathbf{r})$ For a system of non-interacting electrons in an effective potential $v_{eff}(\mathbf{r})$ added to the nuclear potential $v_{nuc}(\mathbf{r})$, we have (dropping the V suffix on the energy functional)

$$E'[\rho_0(\mathbf{r})] = T[\rho_0(\mathbf{r})] + \int \mathrm{d}^3\mathbf{r}\rho_0(\mathbf{r})v_{nuc}(\mathbf{r}) + \int \mathrm{d}^3\mathbf{r}\rho_0(\mathbf{r})v_{eff}(\mathbf{r})(\mathbf{r}) \tag{1 10}$$

where $E'[\rho_0(\mathbf{r})]$ is the energy functional of the system of non-interacting electrons and $T[\rho_0(\mathbf{r})]$ the kinetic energy functional of the same system of non-interacting electrons For the energy functional of the system of interacting electrons we write

$$E[\rho_0(\mathbf{r})] = T[\rho_0(\mathbf{r})] + \int \mathrm{d}^3\mathbf{r}\rho_0(\mathbf{r})v_{nuc}(\mathbf{r}) + \frac{1}{2}\int \mathrm{d}^3\mathbf{r}\int \mathrm{d}^3\mathbf{r}'\rho_0(\mathbf{r})\frac{1}{|\mathbf{r}-\mathbf{r}'|}\rho_0(\mathbf{r}') + E_{xc}[\rho_0(\mathbf{r})] \tag{1 11}$$

The first term is the kinetic energy functional of the non-interacting electron gas The difference between the kinetic energy of the interacting and non-interacting electron gas, is put into E_{xc} The second term is the electron-nucleus interaction, the third term the Hartree potential energy and the fourth term, called the exchange-correlation energy contains everything else We can now use (1 10) to write

$$T[\rho_0(\mathbf{r})] + \int \mathrm{d}^3\mathbf{r}\rho_0(\mathbf{r})v_{nuc}(\mathbf{r}) = E'[\rho_0(\mathbf{r})] - \int \mathrm{d}^3\mathbf{r}\rho_0(\mathbf{r})v_{eff}(\mathbf{r}) \tag{1 12}$$

Substituting (1.12) into (1.11) we get:

$$E[\rho_0(\mathbf{r})] = E'[\rho_0(\mathbf{r})] - \int \mathrm{d}^3\mathbf{r}\rho_0(\mathbf{r})v_{\mathit{eff}}(\mathbf{r}) + \tfrac{1}{2}\int \mathrm{d}^3\mathbf{r}\int \mathrm{d}^3\mathbf{r}'\rho_0(\mathbf{r})\frac{1}{|\mathbf{r}-\mathbf{r}'|}\rho_0(\mathbf{r}') + E_{xc}[\rho_0(\mathbf{r})] \quad (1.13)$$

Since we want an expression for $v_{\mathit{eff}}(\mathbf{r})$, we take the derivative of (1.13) with respect to variations in $\rho_0(\mathbf{r})$:

$$\frac{\partial E}{\partial \rho_0(\mathbf{r})} = \frac{\partial E'}{\partial \rho_0(\mathbf{r})} - v_{\mathit{eff}}(\mathbf{r}) + \int \mathrm{d}^3\mathbf{r}'\frac{1}{|\mathbf{r}-\mathbf{r}'|}\rho_0(\mathbf{r}') + \frac{\partial E_{xc}}{\partial \rho_0(\mathbf{r})} \quad (1.14)$$

Since E and E' are both energies which are stationary with respect to changes in $\rho_0(\mathbf{r})$ the left-hand-side and the first term on the right-hand-side of (1.14) are zero. This gives us the following expression for $v_{\mathit{eff}}(\mathbf{r})$:

$$v_{\mathit{eff}}(\mathbf{r}) = \int \mathrm{d}^3\mathbf{r}'\frac{1}{|\mathbf{r}-\mathbf{r}'|}\rho_0(\mathbf{r}') + \frac{\partial E_{xc}}{\partial \rho_0(\mathbf{r})} \quad (1.15)$$

where the first term in (1.15) is the Hartree potential $V_H(\mathbf{r})$. The second term we will call the exchange-correlation potential $V_{xc}(\mathbf{r})$.

To get from (1.13) to an equation for the system of interacting electrons, we have to get expressions for $E'[\rho_0(\mathbf{r})]$ and $\rho_0(\mathbf{r})$, so we can eliminate E' and $v_{\mathit{eff}}(\mathbf{r})$ from (1.13). We do this by solving the single-particle Schrödinger equation for the non-interacting electrons (the Kohn-Sham equation), to get the ground-state energy and density. The Kohn-Sham equation looks like:

$$\left[-\tfrac{1}{2}\nabla^2 + v_{nuc}(\mathbf{r}) + V_H(\mathbf{r}) + V_{xc}(\mathbf{r})\right]\psi_i(\mathbf{r}) = \epsilon_i\psi_i(\mathbf{r}) \quad (1.16)$$

where $\psi_i(\mathbf{r})$ are single-particle eigenstates and ϵ_i the corresponding eigenvalues. We use this to give us the following results:

$$E' = \sum_i^{occ}\epsilon_i \quad (1.17a)$$

$$\rho_0(\mathbf{r}) = \sum_i^{occ}|\psi_i(\mathbf{r})|^2 \quad (1.17b)$$

Substituting this into (1.13), we get the energy functional for the ground-state energy of the interacting electron system:

$$E[\rho_0(\mathbf{r})] = \sum_i \epsilon_i - \tfrac{1}{2}\int \mathrm{d}^3\mathbf{r}V_H(\mathbf{r})\rho_0(\mathbf{r}) - \int \mathrm{d}^3\mathbf{r}V_{xc}(\mathbf{r})\rho_0(\mathbf{r}) + E_{xc}[\rho_0(\mathbf{r})] \quad (1.18)$$

We now have reduced the original problem with N electrons to a one-particle problem from which we can find the ground-state energy and charge-density of a system of interacting electrons. So where did we lose the many-body (10^{23} electrons) problem? Unfortunately we didn't. The many-body aspect of the system has been hidden in the exchange-correlation energy (and therefore potential). Its functional dependence on $\rho_0(\mathbf{r})$ has to be known to solve (1.18).

So why did we derive this density-functional-theory? It appears we can't use it since we do not know the functional form of E_{xc}. The answer to this is the subject of the next section, where we approximate E_{xc}.

1.2.4 Local density approximation

Since we do not know the exact functional dependence of E_{xc} on $\rho_0(\mathbf{r})$, we try an approximate form:

$$E_{xc}[\rho_0(\mathbf{r})] \approx \int \mathrm{d}^3\mathbf{r}\rho_0(\mathbf{r})\epsilon_{xc}(\rho_0(\mathbf{r})) \tag{1.19}$$

where $\epsilon_{xc}(\rho_0(\mathbf{r}))$ is the exchange-correlation energy of an infinite, homogeneous electron-gas with a density equal to the *local density* $\rho_0(\mathbf{r})$.

In this approximation, the exchange-correlation potential is given by:

$$V_{xc}(\mathbf{r}) = \frac{\mathrm{d}}{\mathrm{d}\rho}\left(\rho\epsilon_{xc}(\rho)\right)\Big|_{\rho=\rho_0(\mathbf{r})} \tag{1.20}$$

The exchange-correlation energy for the uniform electron-gas is well-known within certain limits. There are several ways of calculating it, most of which can be used in most modern codes for doing electronic structure calculations.

It is clear that the local density approximation (LDA), as this approximation is called works rather well for systems where the density is slowly varying. In practice it works well even for atoms, molecules and surfaces.

Now we have used several theories and approximations, to bring our original problem with many particles (of the order of 10^{23}) down to an one-particle problem. How do we elegantly solve this one-particle problem? One way of solving for the one-particle density is using Green functions. This is the topic of the next section.

1.2.5 Green functions

Green functions have become accepted tools in most areas of mathematical physics. Whereas the wavefunction sometimes contains a lot of extra unnecessary information, the Green function will give you all the quantities you need. And also it is very easy to impose boundary conditions through the Green functions. Since the boundary conditions on the Kohn-Sham equation will become very important later on in the method, it seems useful to perform our calculations using Green functions. Because the Kohn-Sham equation is a single particle equation, we will concentrate in this section on the single-particle Green function.

We define the Green function $G(\mathbf{r},\mathbf{r}';\mathcal{E})$ by:

$$\left[-\tfrac{1}{2}\nabla^2 + V(\mathbf{r}) - \mathcal{E}\right] G(\mathbf{r},\mathbf{r}';\mathcal{E}) = \delta(\mathbf{r}-\mathbf{r}') \tag{1.21}$$

where $\mathcal{E}$ is a complex quantity, which is essentially an energy analytically extended into the complex plane. $V(\mathbf{r})$ contains all contributions to the potential as discussed before.

We see that the Green function is a non-local energy dependent quantity. So how do we solve for it? Let us write (1.21) in operator form:

$$[H - \mathcal{E}]\, G = 1 \tag{1.22}$$

Then we could also write:

$$G = [H - \mathcal{E}]^{-1} \tag{1.23}$$

This is actually the approach most people use to solve for the Green function. You choose a basis. Then you calculate the Hamiltonian (and for a non-orthonormal basis the overlap) matrix. Then finding the Green function matrix becomes a matter of inverting a matrix.

Once we have the Green function, we need a new definition for (1.17b), since we do not have wavefunctions in our method. We can use the following definition:

$$\rho(\mathbf{r}) = \frac{1}{\pi}\mathrm{Im}\int_{-\infty}^{E_F} \mathrm{d}\mathcal{E}\, G(\mathbf{r},\mathbf{r},\mathcal{E}) \tag{1.24}$$

where E_F is the energy of the highest occupied state, the Fermi energy. It should become clear from this why we do not evaluate G at real energies, because then this integral will have problems with the eigenvalues of H, which are poles of G. It turns out to be very useful to deform the integral in (1.24) to a half circle from some negative energy below all occupied states to the Fermi energy.

Also the local density of states (LDOS) can be derived from the Green function, when we set $\mathcal{E} = E + i\epsilon$, where E and ϵ are both real:

$$\sigma(\mathbf{r}, E) = \lim_{\epsilon\to 0}\frac{1}{\pi}\mathrm{Im}G(\mathbf{r},\mathbf{r}; E + i\epsilon) \tag{1.25}$$

We now have reduced the problem to a single-particle model, we know how to solve for the Green function and get physical properties from it. But we still have to solve in all space. We have to find a method for limiting $\mathbf{r}$ to a volume we can handle.

1.2.6 Using the symmetry of the system

To be able to use our single particle equation, we often limit the systems we can handle to systems with a certain symmetry. The most useful symmetry in this aspect is the translational symmetry. When we say that a material is invariant under a certain translation $\mathbf{r}_t$, we mean that the environment around a point in space $\mathbf{r}$ is the same as around the point $\mathbf{r} + \mathbf{r}_t$. So when we look at the potential, we get:

$$V(\mathbf{r} + \mathbf{r}_t) = V(\mathbf{r}) \tag{1.26}$$

In most cases we want to study crystals and we have a set of vectors $\mathbf{r}_t$, which spans the entire system, namely the vectors of the crystal lattice.

When we have a system with such a translational symmetry, we can use Bloch's theorem[2], which tells us:

$$\psi_{\mathbf{k}}(\mathbf{r} + \mathbf{r}_t) = e^{i\mathbf{k}\cdot\mathbf{r}_t}\psi_{\mathbf{k}}(\mathbf{r}) \tag{1.27}$$

where $\mathbf{k}$ is a wavevector in reciprocal space. We see that essentially we only have to solve for $\psi_{\mathbf{k}}(\mathbf{r})$ in the unit cell of the lattice and this will give us the behaviour of that wavefunction everywhere. We have now an extra index on the wavefunction; this is the price we have to pay for limiting the space in which we solve (1.21). Normally one has to integrate over $\mathbf{k}$ in the first Brillouin zone to get physical properties such as the charge density and the total density of states.

With all the approximations and theorems we have used up to now, we are able to perform calculations for crystals with perfect translational symmetry. However, normally we will also be interested in systems with "broken symmetries". There are some ways to handle symmetry-breakings that are now well established. These will be discussed in the next section. A relatively new way to handle locally broken symmetries will be discussed in section 1.4

1.3 Tricks often used for dealing with broken symmetries

Before trying to outline some tricks for handling them, it would be wise to first clarify what sort of symmetry-breakings we are talking about.

1.3.1 Ways to break symmetry

The starting point in this section will be the perfectly translationally symmetric crystal. Among the many ways to destroy the symmetry of the crystal, we will concentrate on two here, the bulk substitutional impurity and the surface.

Bulk substitutional impurity

To clarify what is meant by a bulk substitutional impurity, I have sketched two pictures in figure 1.1: the picture on the left is a two-dimensional perfect "crystal" made up of black circles (assume this sketch extends to infinity in all four relevant directions). We now substitute one black circle with a slightly larger gray one, to get the right-hand picture. This is what we call a substitutional impurity. It is clear that the symmetry-breaking in this case is 'local'. In

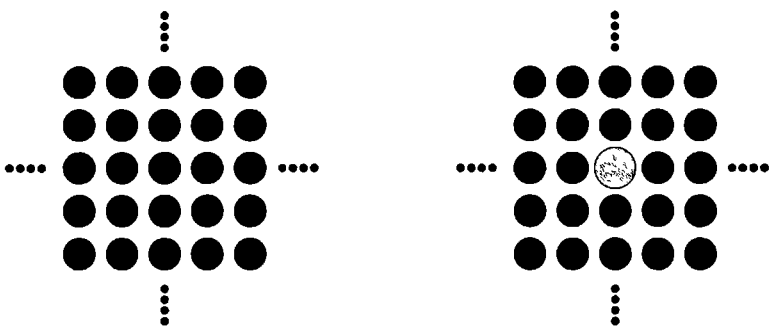


Figure 1.1: *Left: sketch of a perfect bulk two-dimensional "crystal" of black circles. Right: sketch of a substitutional impurity in this bulk system.*

the case of the surface this is not apparent at first sight.

A surface as a (locally) broken symmetry

Another way to break symmetry, quite drastically this time, is to split the crystal in two to make a surface. This is shown in figure 1.2. The figure on the left is again the perfect crystal of black circles, extending in all four directions. The figure on the right is the surface, where it is important to note that the crystal now only extends to the left, right, and bottom. In the case of the surface it is not completely clear that the symmetry is only broken locally. But since we can use Bloch's theorem in the direction parallel to the surface, we can limit the extent of the surface in the parallel dimension. We can then see that the change is localized in the perpendicular direction, around the interface between the black circles and 'empty space'.

We will now look at some standard tricks to handle these broken symmetries.

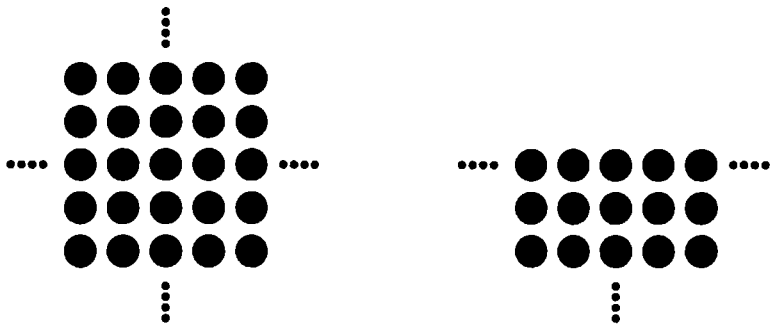


Figure 1.2: *Left: sketch of a perfect bulk two-dimensional "crystal" of black circles. Right: sketch of a surface, created by splitting the bulk in two.*

1.3.2 How broken symmetries are treated in current methods

The standard way to deal with the broken symmetry has been to restore the symmetry, by constructing a new crystal. This new crystal has a large unit-cell, which contains the symmetry-breaking entity. These new, large unit-cells are usually called supercells. Examples for the bulk impurity and surface are shown in figure 1.3. It now becomes quite simple to treat these systems which originally had a lower symmetry, because we have restored symmetry, and can therefore again use the machinery that has been refined for 'perfect' crystals.

It is clear that care must be taken to choose the supercell big enough. For example in the case of a surface, the supercell must be so big that the middle of the slab of atoms is essentially 'bulk-like' and the surfaces should not interact. This means that both the region of vacuum and the number of layers in the slab have to be reasonably big (typically 7-13 layers are used).

The method of supercells is very useful when one is interested in local physical quantities like the charge density, or work-function. But the description of the density of states in the surface problem for example, is not treated properly. Instead of a bulk continuum, we get a finite collection of discrete states. It then becomes a rather complicated process to separate out surface-localized states from the bulk.

1.4 A new way to deal with locally broken symmetries

The problem with the density of states motivates us to develop an "embedding" approach for solving the Schrödinger equation for the real system and not one with artificially restored symmetry. Other advantages will emerge as we shall see. In this introduction I will concentrate on the the case of a surface, and I will merely outline how the embedding approach works. Later in chapter 2 a more detailed description will be given.

In the case of the surface, we expressed a need for the true bulk continuum. It turns out that one can only get a continuum of states if one has a true (semi-)infinite system. So, how do we go from the situation in the picture on the right of figure 1.2 to a system which is bounded in space, but which is able to couple to a semi-infinite substrate.

Let us look at the following situation. We split space into three regions: a semi-infinite

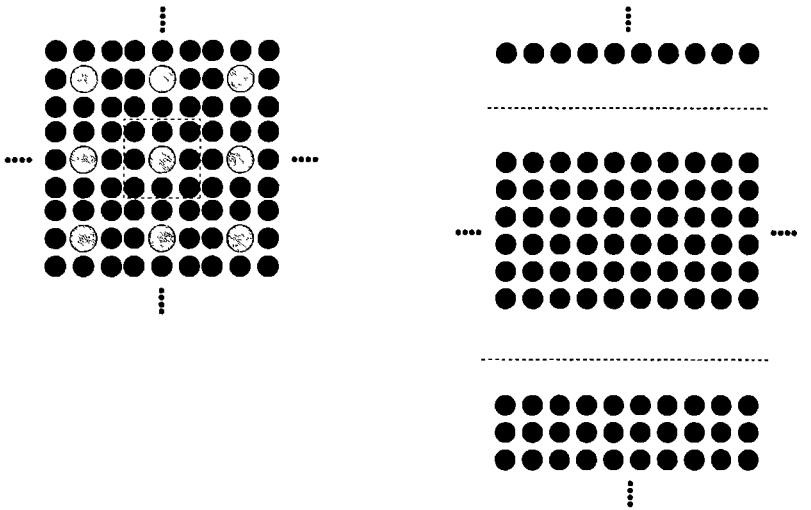


Figure 1.3: *Left: sketch of the supercell approach for a bulk impurity. The unit cell has been indicated by a dashed line. Right: sketch of a surface, repeated in the direction perpendicular to the surface. The repeat cell is indicated by the two dashed lines*

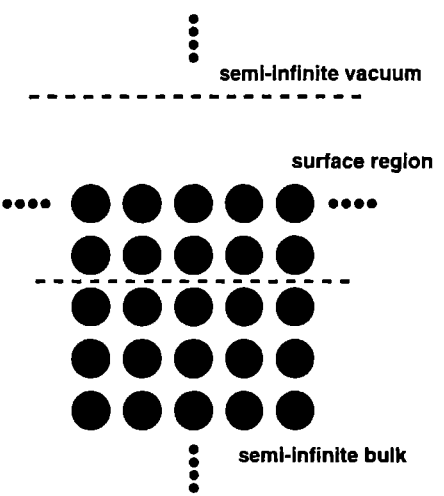


Figure 1.4: *Space is split in 3 regions. On top the semi-infinite vacuum, which stretches to infinity in the up, left and right directions. On the bottom the semi-infinite bulk, which stretches to infinity in the down, left and right directions. And in the middle the surface region, which stretches to infinity in the left and right directions.*

bulk crystal, a semi-infinite vacuum and a system around the surface, see figure 1.4. The trick is to somehow solve the single particle equation only in the surface region, and couple the continua on both sides to this system. It turns out that once we know the Green function of the semi-infinite substrates (bulk and vacuum), we can construct a non-local energy-dependent potential on the planes separating the surface region from the substrates. We add this potential, to be called the embedding potential to the potential already present in the Kohn-Sham equation.

When we now solve this equation, we not only get a good representation of what happens at the surface, but we also get a continuum of states because of the substrates.

The rest of this thesis will concentrate on the exact derivation of this new trick. Another part of this thesis is the implementation of this trick, together with some smaller tricks to enhance efficiency of the method. Also, the results of some interesting applications of the method will be discussed.

Part I

Theory

Chapter 2

Generalised Embedding

In this chapter the theory of the embedding method will be derived for a general geometry. It will be demonstrated how this enables us to handle most locally broken symmetries, once we can solve the unperturbed system. But first I shall try to explain the importance of boundary conditions and why embedding works in most cases. An insider's point of view of embedding is presented in non-physics terms in appendix A.

2.1 The importance of boundary conditions

To make clear that boundary conditions are important with respect to some non-local properties, I want to make the following observations:

- We have to distinguish between wavefunction properties and integrated properties. By integrated properties I mean properties integrated over energy. For example, individual wavefunctions are very sensitive to local perturbations, whereas integrated quantities such as charge density, potential or energy are not. This is the reason that supercell and slab calculations perform so well when concentrating on the behaviour of those integrated properties.

- However, details of the density of states or band structure (spectrum) can be very sensitive to local perturbations. In particular discrete states are very sensitive, whereas continuum states will be less sensitive, because these are in a sense again a sort of integrated quantity. Even though individual states in the continuum may vary, the total continuum will remain more or less the same when the perturbation is local.

- Especially in metals the range of a potential perturbation is rather short, as a result of both screening and the fact that the charge density is an integrated quantity. In semiconductors and insulators the long range fields are rather smoothly varying.

From this it is hopefully clear that when we are interested in more than just integrated quantities, we will have to handle the boundary conditions with care. So if we want to calculate a surface density of states, to examine surface states, and we want to use a geometry as shown in figure 1.4, we will have to construct precise boundary conditions on the planes in figure 1.4.

These boundary conditions on either the wavefunction or the Green function, will be incorporated in the Schrödinger equation through a reflectivity. So in a sense we want to

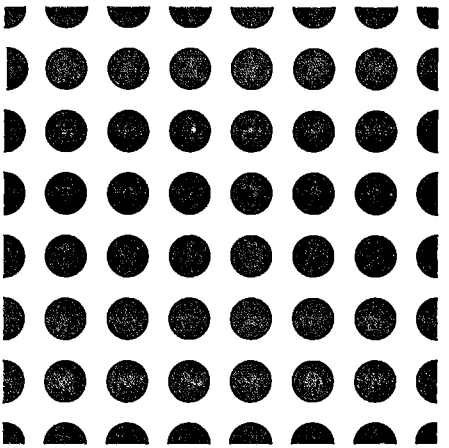


Figure 2.1: *A schematic picture of the bulk system.*

trick the electrons into thinking that behind the plane on which we construct the boundary condition, the physical world extends onwards. We will see later that we do this by specifying a generalised logarithmic derivative for the wavefunctions or Green function on our boundary-condition-plane. This generalised logarithmic derivative is called the embedding potential. The embedding potential itself is non-local, which means it not only depends on the local coordinate $\mathbf{r}$ but also on the other points in space $\mathbf{r}'$. It is also energy-dependent, in other words it is not an integrated property.

The boundary-condition-plane or surface we call the embedding surface. This embedding surface has to be a closed surface around the localised symmetry-breaking entity. We can call this surface a 'box'. We choose the name box, because in the case of realistic systems this surface will be box-like, because we want it to cut through as few atoms as possible.

2.2 Embedding in a box

I will now explain the embedding approach, using the bulk impurity as an example.

As the first step in any calculation involving embedding, we have to calculate the Green function of the system into which we want to embed. The unperturbed system in the case of a bulk impurity is the normal bulk. A cut through the clean unperturbed bulk system is shown in figure 2.1.

We now have to choose our embedding surface or box. The box has to be large enough to include the impurity and some of the surrounding atoms, so that it contains the whole region where the potential is significantly perturbed. We then construct the embedding potential on the surface of the box, from the properties of the perfect host material. We may now solve the Schrödinger equation inside the box, the embedding potential providing the required boundary conditions on the wavefunctions. Replacing an atom in the box by an impurity (figure 2.2), and subsequently going to self-consistency, does not affect the embedding potential which takes care of all the effects of the substrate on the wavefunctions. This approach is quite different from the supercell approach, in that we apply boundary conditions

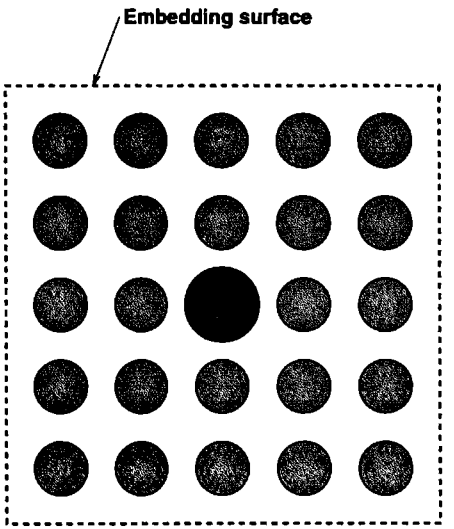


Figure 2.2: *The box in which we perform our calculations. The centre atom has been replaced by the impurity.*

on the wavefunction, which should make it behave as if there is an infinite bulk outside the embedding surface. In a supercell the wavefunction is artificially made periodic, which leads to discrete states (the so-called spaghetti band-structure).

When we perform the calculation inside the box we can see that, for example, the electron density around the impurity will be higher than around the original atom in the clean bulk. Up to now we have talked about embedding in pictures. In the next section we will start to look at embedding in terms of the Schrödinger equation and Green functions.

2.3 Embedding in equations

In this section we wish to describe embedding in very general terms, and end up with the embedding equation[3]. The first step is to split the physical world into two parts. One part which we will call the outside world, is of infinite size, and is schematically indicated in figure 2.3 as region 2. The other part, finite in size, is the system in which we are interested. This region is indicated as 1. The two regions are separated by a closed surface, called S_1.

In the situation we want to reach we solve the Schrödinger equation explicitly only in region 1. In this case the entire *outside* (i.e. region 2) is taken into account by an embedding potential on surface S_1. First we take a trial wavefunction ψ in all space, defined as:

$$\psi(\mathbf{r}) = \begin{cases} \phi_1(\mathbf{r}), & \text{for } \mathbf{r} \text{ in region 1,} \\ \phi_2(\mathbf{r}), & \text{for } \mathbf{r} \text{ in region 2} \end{cases} \tag{2.1}$$

where ϕ_1 and ϕ_2 match in amplitude over surface S_1 and $\phi_2(\mathbf{r})$, the trial wavefunction in region 2, is a solution of the Schrödinger equation in this region at energy ϵ. We now calculate

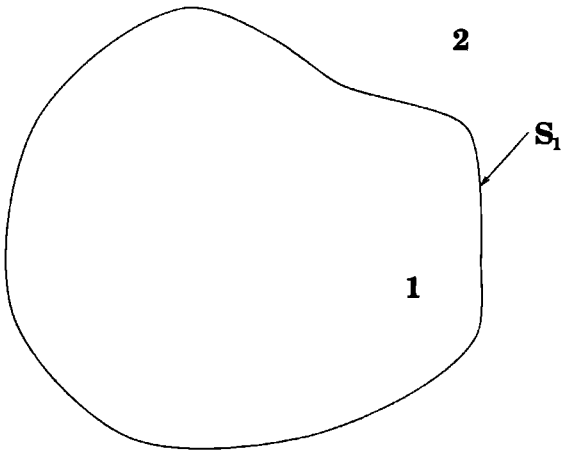


Figure 2.3: *Schematic picture of how we partition space.*

the expectation value of the Hamiltonian in all space with respect to this trial function.

$$E = \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} = \frac{\int_1 \mathrm{d}^3\mathbf{r}\phi_1^* H\phi_1 + \int_2 \mathrm{d}^3\mathbf{r}\phi_2^* H\phi_2 + \frac{1}{2}\int_{S_1} \mathrm{d}^2\mathbf{r}_S\phi_1^*\frac{\partial\phi_1}{\partial n_S} - \frac{1}{2}\int_{S_1} \mathrm{d}^2\mathbf{r}_S\phi_2^*\frac{\partial\phi_2}{\partial n_S}}{\int_1 \mathrm{d}^3\mathbf{r}|\phi_1|^2 + \int_2 \mathrm{d}^3\mathbf{r}|\phi_2|^2} \tag{2.2}$$

Since we did not specify that ϕ_1 and ϕ_2 match in derivative over S_1, we get surface terms from the kinetic energy associated with the discontinuity of $\frac{\partial\phi_1}{\partial n_S}$ and $\frac{\partial\phi_2}{\partial n_S}$.

Important note: In the remainder of this thesis the normal derivative will show up more often. This normal *always* points **into** the system we are embedding in.

In this case the normal points from region 1 into region 2. Using the fact that $\phi_2(\mathbf{r})$ is a solution of the Schrödinger equation, we obtain:

$$E = \frac{\int_1 \mathrm{d}^3\mathbf{r}\phi_1^* H\phi_1 + \epsilon\int_2 \mathrm{d}^3\mathbf{r}|\phi_2|^2 + \frac{1}{2}\int_{S_1} \mathrm{d}^2\mathbf{r}_S\phi_1^*\frac{\partial\phi_1}{\partial n_S} - \frac{1}{2}\int_{S_1} \mathrm{d}^2\mathbf{r}_S\phi_2^*\frac{\partial\phi_2}{\partial n_S}}{\int_1 \mathrm{d}^3\mathbf{r}|\phi_1|^2 + \int_2 \mathrm{d}^3\mathbf{r}|\phi_2|^2} \tag{2.3}$$

Since we do not really want to solve the Schrödinger equation explicitly in the *outside* region, we want to get rid of ϕ_2 in this equation. This can be done if we know the Green function for the outside region $G_0(\mathbf{r},\mathbf{r}')$. As an extra condition on the Green function we demand the following boundary condition on surface S_1:

$$\frac{\partial G_0(\mathbf{r}_S,\mathbf{r}')}{\partial n_S} = 0 \tag{2.4}$$

We are free to do this, by adding solutions of the homogenous Schrödinger equation in region 2 to the Green function. Now let us define the *surface*-inverse of the Green function by:

$$\int_{S_1} \mathrm{d}^2\mathbf{r}_S' \widehat{G_0}^{-1}(\mathbf{r}_S, \mathbf{r}_S') G_0(\mathbf{r}_S', \mathbf{r}_S'') = \delta(\mathbf{r}_S - \mathbf{r}_S'') \tag{2.5}$$

Surface inverses will be indicated by both a hat ($\widehat{\ }$) and an exponent -1. We can use Green's theorem in region 2 to relate the wavefunction to the normal derivative in the following way (see chapter 3 of [4]):

$$\frac{\partial \phi_2(\mathbf{r}_S)}{\partial n_S} = -2 \int_{S_1} \mathrm{d}^2\mathbf{r}_S' \widehat{G_0}^{-1}(\mathbf{r}_S, \mathbf{r}_S') \phi_2(\mathbf{r}_S') \tag{2.6}$$

From this it is clear that $\widehat{G}_0^{-1}$ is related to a logarithmic derivative of the wavefunction. When we use the fact that ϕ_1 and ϕ_2 match in amplitude over S_1, with the following relationship between the normalization integral in region 2 and the energy derivative of the surface-inverse Green function[3]:

$$\int_2 \mathrm{d}^3\mathbf{r} |\phi_2(\mathbf{r})|^2 = -\int_{S_1} \mathrm{d}^2\mathbf{r}_S \int_{S_1} \mathrm{d}^2\mathbf{r}_S' \phi_1^*(\mathbf{r}_S) \frac{\partial \widehat{G_0}^{-1}(\mathbf{r}_S, \mathbf{r}_S')}{\partial E} \phi_1(\mathbf{r}_S') \tag{2.7}$$

we find:

$$E = \frac{\int_1 \mathrm{d}^3\mathbf{r} \phi_1^* H \phi_1 + \frac{1}{2}\int_{S_1} \mathrm{d}^2\mathbf{r}_S \phi_1^* \frac{\partial \phi_1}{\partial n_S} + \int_{S_1} \mathrm{d}^2\mathbf{r}_S \int_{S_1} \mathrm{d}^2\mathbf{r}_S' \phi_1^*(\mathbf{r}_S) \left[\widehat{G_0}^{-1} - \epsilon \frac{\partial \widehat{G_0}^{-1}}{\partial E} \right] \phi_1(\mathbf{r}_S')}{\int_1 \mathrm{d}^3\mathbf{r} |\phi_1(\mathbf{r})|^2 - \int_{S_1} \mathrm{d}^2\mathbf{r}_S \int_{S_1} \mathrm{d}^2\mathbf{r}_S' \phi_1^*(\mathbf{r}_S) \frac{\partial \widehat{G_0}^{-1}}{\partial E} \phi_1(\mathbf{r}_S')} \tag{2.8}$$

which no longer contains any reference to ϕ_2. So we have expressed the problem entirely in terms of ϕ_1, the trial function in region 1.

It turns out that we have put all the information about the *outside* region into the *surface-inverse* of the outside Green function ($\widehat{G_0}^{-1}$), evaluated at energy ϵ. Applying the variational principle to (2.8) we can derive the Schrödinger-like equation which we have to solve in region 1 in order to determine ϕ_1. This equation can be written as:

$$\left[H + \frac{1}{2}\frac{\partial}{\partial n_S}\delta(\mathbf{r} - \mathbf{r}_S) + \left(\widehat{G_0}^{-1} + (E - \epsilon)\frac{\partial \widehat{G_0}^{-1}}{\partial E} \right) \delta(\mathbf{r} - \mathbf{r}_S)\delta(\mathbf{r}' - \mathbf{r}_S') \right] \phi_1(\mathbf{r}) = E\phi_1(\mathbf{r}) \tag{2.9}$$

where $H\phi = E\phi$ is the *normal* Schrödinger equation in region 1. All terms in (2.9) except the Hamiltonian H are surface terms, i.e. the operators only exist on the surface S_1, as indicated by the δ-function. The term $\dfrac{\partial}{\partial n_S}$ is in there to make the Hamiltonian hermitian (because the kinetic energy is not hermitian in an enclosed region). Remember, the normal points out of the region of interest, in this case out of region 1. $\widehat{G_0}^{-1}$ is what we will call the *embedding potential*. The last term of the LHS is a correction term that we have to put in when $\widehat{G_0}^{-1}$

is not evaluated at energy E. If we now make sure that we always evaluate the embedding potential at the energy, E, at which we are working, we obtain the following equation:

$$\left[-\tfrac{1}{2}\nabla_{\mathbf{r}}^2 + V'(\mathbf{r}) + \tfrac{1}{2}\frac{\partial}{\partial n_S}\delta(\mathbf{r}-\mathbf{r}_S) + \widehat{G_0}^{-1}(E)\delta(\mathbf{r}-\mathbf{r}_S)\delta(\mathbf{r}'-\mathbf{r}'_S)\right]\phi_1(\mathbf{r}) = E\phi_1(\mathbf{r}) \tag{2.10}$$

where we have replaced H by $-\frac{1}{2}\nabla_{\mathbf{r}}^2 + V'(\mathbf{r})$. Equation (2.10) is what from now on will be called the basic embedding equation. $V'(\mathbf{r})$ is the perturbed potential in region 1.

Before concentrating on layered systems in the rest of this thesis, I will first discuss the box-embedding method.

2.4 How to construct the embedding potential

In the previous sections it was assumed that we could readily find the embedding potential. In section 2.3 a definition for the embedding potential was given, but this definition depended on a known Green function of the outside system, with zero normal derivative on the embedding surface. Even if a Green function of the outside system is known, it doesn't always have zero normal derivative. How to handle that will be discussed later.

The situation most often occurring, is that we know the Green function of an unperturbed system, i.e. of a system $1+2$, where in the region 1 we now have the unperturbed potential $V(\mathbf{r})$. In analogy to the bulk impurity discussed earlier, 1 is now the box filled with normal bulk atoms. So we have two potentials, the **unperturbed** potential $V(\mathbf{r})$ (which throughout region 1 we designate V_1) and the **perturbed** potential $V'(\mathbf{r})$ (which throughout region 1 we designate V_1'). We will use the partitioning of space as shown in figure 2.4. So we want to go from G_{1+2}, the Green function for $1+2$ to G_2 the Green function for 2 only, with zero normal derivative on S_1. We will assume that we have an embedding potential on the external surface S_2. This may be a surface at infinity, where we can choose a zero embedding potential. We are free to choose a zero embedding potential, because of the boundary condition at infinity that the wavefunction and its derivatives are zero. The Green function for the entire $1+2$ system is given by:

$$G_{1+2} = (H_{1+2} - EI_{1+2})^{-1} \tag{2.11}$$

where I indicates the unity operator. From here on I shall indicate surface terms in the Hamiltonian by a line behind the term with as subscript the name of the surface. To find the Green function for 2 with zero normal derivative on S_1 we write down an embedded Hamiltonian for region 2, setting the embedding potential on S_1 to be zero:

$$H_2 = T_2 + V_2 + \tfrac{1}{2}\frac{\partial}{\partial n_S}\Big|_{S_1} + \tfrac{1}{2}\frac{\partial}{\partial n_S}\Big|_{S_2} + \widehat{G_0}^{-1}\Big|_{S_2} \tag{2.12}$$

Again I want to stress the importance of the direction of the normal: here the normal on surface S_1 points into region 1 (we are embedding region 2 this time!), and the normal on S_2 points out of 2. Defining G_2 using:

$$G_2 = (H_2 - EI_2)^{-1}, \tag{2.13}$$

we ensure:

$$\frac{\partial}{\partial n_S}\Big|_{S_1} G_2(\mathbf{r}, \mathbf{r}_S) = 0, \tag{2.14}$$

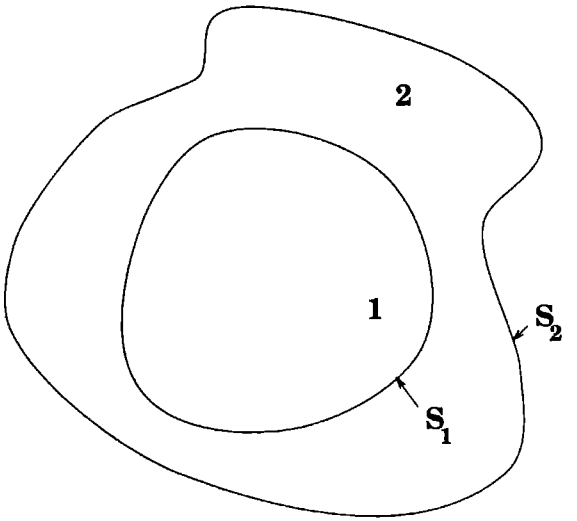


Figure 2.4: *Partitioning of space for finding the embedding potential in S_1 from the Green function in 1+2.*

because setting the embedding potential on a surface to zero implies that the Green function that is embedded will have zero normal derivative on that surface. We now want to write G_{1+2} in such a way, that we can distil an expression for G_2 from it. We do this in the following manner. We know that G_{1+2} is defined by (2.11), where H_{1+2} is given by:

$$H_{1+2} = T_{1+2} + V_{1+2} + \frac{1}{2}\frac{\partial}{\partial n_S}\Big|_{S_2} + \widehat{G_0}^{-1}\Big|_{S_2} \tag{2.15}$$

This leads to the following expression for the inverse[1] of G_{1+2}

$$G_{1+2}^{-1} = T_{1+2} + V_{1+2} + \frac{1}{2}\frac{\partial}{\partial n_S}\Big|_{S_2} + \widehat{G_0}^{-1}\Big|_{S_2} - EI_{1+2} \tag{2.16}$$

We now use the fact that:

$$T_{1+2} = T_1 + T_2 \tag{2.17a}$$
$$V_{1+2} = V_1 + V_2 \tag{2.17b}$$
$$I_{1+2} = I_1 + I_2 \tag{2.17c}$$

which means we can write (2.16) as:

$$G_{1+2}^{-1} = T_1 + V_1 - EI_1 + T_2 + V_2 - EI_2 + \frac{1}{2}\frac{\partial}{\partial n_S}\Big|_{S_2} + \widehat{G_0}^{-1}\Big|_{S_2} \tag{2.18}$$

We now combine (2.12), (2.13) and (2.18) to give us an expression for the inverse of G_2:

$$G_2^{-1} = G_{1+2}^{-1} - T_1 - V_1 + EI_1 + \frac{1}{2}\frac{\partial}{\partial n_S}\Big|_{S_1} \tag{2.19}$$

[1]This is a normal operator inverse, i.e. not a surface-inverse.

We can then invert G_2^{-1} to give the Green function in region 2 which obeys (2.14), and therefore when *surface*-inverted will give us the embedding potential for region 1 on S_1.

The expression in (2.19) seems to make sense. Since we want a Green function in region 2 with a certain boundary condition on S_1 (zero normal derivative), we wouldn't want the Green function constrained in region 1. Thus it is logical to invert G_{1+2} to get a Hamiltonian, and then subtract all terms coming from region 1, thereby unconstraining the Green function in that region. Then we can add the normal derivative term to ensure that the resulting Green function obeys our boundary condition.

I want to stress that in all the derivations above, the region 1 is part of the unperturbed system, i.e. V_1 is the unperturbed potential integrated over region 1.

2.4.1 Green's theorem approach to finding the embedding potential

So now we are able to go from a Green function for the 'clean' system to an embedding potential. An alternative way of getting the embedding potential is to start from a Green function for the outer region which has arbitrary normal derivative on S_1 and changing that to the required Green function with zero normal derivative, using Green's theorem arguments.

The geometry most often occurring for this case is the one shown in figure 2.3. What we have through some other means, is a Green function in region 2, with arbitrary boundary condition on S_1. We will call this Green function G. We want a Green function for region 2 with zero normal derivative on S_1, which we will call G_0. Since both Green functions are solutions inside region 2, we can write:

$$(H_2 - E)\, G_0(\mathbf{r}, \mathbf{r}'') = \delta(\mathbf{r} - \mathbf{r}'') \tag{2.20a}$$

$$(H_2 - E)\, G(\mathbf{r}'', \mathbf{r}') = \delta(\mathbf{r}'' - \mathbf{r}') \tag{2.20b}$$

We multiply (2.20a) with $G(\mathbf{r}'', \mathbf{r}')$ and (2.20b) with $G_0(\mathbf{r}, \mathbf{r}'')$ and get:

$$G(\mathbf{r}'', \mathbf{r}')(H_2 - E)G_0(\mathbf{r}, \mathbf{r}'') = G(\mathbf{r}'', \mathbf{r}')\delta(\mathbf{r} - \mathbf{r}'') \tag{2.21a}$$

$$G_0(\mathbf{r}, \mathbf{r}'')(H_2 - E)G(\mathbf{r}'', \mathbf{r}') = G_0(\mathbf{r}, \mathbf{r}'')\delta(\mathbf{r}'' - \mathbf{r}') \tag{2.21b}$$

Subtracting these two equations and integrating $\mathbf{r}''$ over region 2, we obtain:

$$-\tfrac{1}{2}\int_2 \mathrm{d}^3\mathbf{r}'' \left[G(\mathbf{r}'', \mathbf{r}')\nabla^2 G_0(\mathbf{r}, \mathbf{r}'') - G_0(\mathbf{r}, \mathbf{r}'')\nabla^2 G(\mathbf{r}'', \mathbf{r}')\right] = \int_2 \mathrm{d}^3\mathbf{r}'' \left[G(\mathbf{r}'', \mathbf{r}')\delta(\mathbf{r} - \mathbf{r}'') - G_0(\mathbf{r}, \mathbf{r}'')\delta(\mathbf{r}'' - \mathbf{r}')\right] \tag{2.22}$$

Using Green's theorem we obtain the following equation:

$$-\tfrac{1}{2}\int_{S_1} \mathrm{d}^2\mathbf{r}_S \left[G(\mathbf{r}_S, \mathbf{r}')\frac{\partial}{\partial n_S} G_0(\mathbf{r}, \mathbf{r}_S) - G_0(\mathbf{r}, \mathbf{r}_S)\frac{\partial}{\partial n_S} G(\mathbf{r}_S, \mathbf{r}')\right] = G(\mathbf{r}, \mathbf{r}') - G_0(\mathbf{r}, \mathbf{r}') \tag{2.23}$$

We now use the fact that G_0 has zero normal derivative on S_1, to give us:

$$G_0(\mathbf{r}, \mathbf{r}') = G(\mathbf{r}, \mathbf{r}') - \tfrac{1}{2}\int_{S_1} \mathrm{d}^2\mathbf{r}_S G_0(\mathbf{r}, \mathbf{r}_S)\frac{\partial}{\partial n_S} G(\mathbf{r}_S, \mathbf{r}') \tag{2.24}$$

We can then use this integral equation to solve for G_0. A surface inversion gives once again the required embedding potential for region 1.

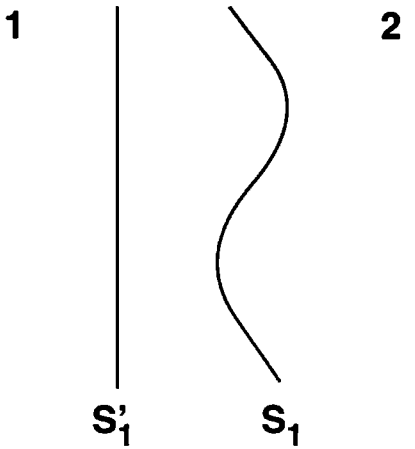


Figure 2.5: *Regions 1 and 2 are separated by the "complicated" embedding surface S_1. We now want to transfer the boundary conditions to the new embedding surface S_1'.*

2.4.2 Solving the Schrödinger equation inside region 1

Once we have the embedding potential on surface S_1, $\widehat{G_2}^{-1}$, we can solve the Schrödinger equation inside region 1, using:

$$G_1 = \left[T_1 + V_1' + \frac{1}{2}\frac{\partial}{\partial n_S}\bigg|_{S_1} + \widehat{G_2}^{-1} - EI_1 \right]^{-1} \tag{2.25}$$

where V_1' is the perturbed potential in region 1, for example the potential of the impurity in a piece of bulk. Now the normal n_S is pointing out of region 1 into region 2.

2.5 Moving the embedding surface

In the description of embedding in previous sections we have assumed some shape for the embedding surface. However, because of implementation limitations, we may want to use a different surface from the surface separating region 1 and 2. In this section I shall discuss whether we can move and reshape the embedding surface and how to go about it.

What we want to accomplish is to transfer the boundary condition on some surface S_1 to another surface S_1' – see figure 2.5. A derivation of this transfer has been presented earlier[5], but here I want to make a little clearer what actually needs to be done.

The situation as used before in this chapter is shown in figure 2.6. This is our starting point. The next step is to put the new embedding surface S_1' in. We now concentrate on the region in between, which we will call Δ. In region Δ we will now work with two different Schrödinger equations, with different potentials, the one belonging to region 1, V_1, and the potential of region 2 extended into region Δ, V_2. The trial function in region 2, Ψ_2, is extended into Δ by integrating the Schrödinger equation in region Δ with potential V_2. In region 1

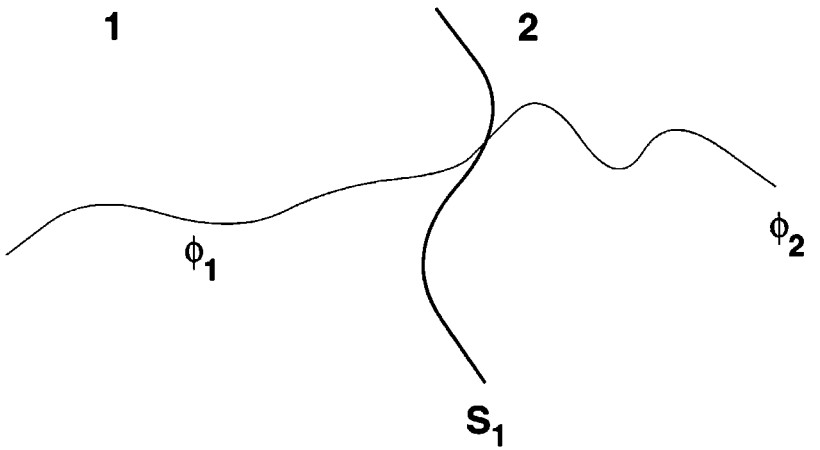


Figure 2.6: *Region 1 and 2 are separated by embedding surface S_1. Trial function ϕ_1 'lives' in region 1 and trial function ϕ_2 in region 2.*

we have a trial function, called Ψ_1, that extends to S_1, because Δ is in a sense part of 1. However, take note, the trial function Ψ_1 feels V_1 in Δ, not V_2. Starting from the logarithmic derivative of the trial function Ψ_1 on S_1, we integrate the Schrödinger equation back with potential V_2 to surface S_1'. This new part of the trial function we call Ψ_1'. The situation now reached is shown in figure 2.7. The matching condition now becomes that Ψ_1' should have the same logarithmic derivative on S_1' as Ψ_2. If we apply this to the situation in figure 2.7, we note that since Ψ_2 and Ψ_1' have the same logarithmic derivative on S_1' and they feel the same potential (V_2) in Δ, the two functions are the same. It then follows that Ψ_1' and Ψ_2 have the same logarithmic derivative on S_1. Since Ψ_1 and Ψ_1' have the same logarithmic derivative on S_1, it follows that Ψ_1 and Ψ_2 have the same logarithmic derivative on S_1, which was the original boundary condition. This final situation is shown in figure 2.8. We see that the boundary condition was imposed at the point A indicated in that figure. So we have transfered the original boundary condition on S_1 to an equivalent boundary condition on S_1'.

We have learned that we can transfer the boundary condition from S_1 to S_1' by integrating the Schrödinger equation back through V_2 in region Δ. A similar argument holds for all different geometries, also when we have a situation where S_1 and S_1' cross in several places and Δ lies not purely in either region 1 or region 2. In practice this procedure means that we will have to subtract from the Hamiltonian both a kinetic energy and a potential term for region Δ.

As an example of this, let us look at a situation which will show up in later chapters, where we want to use an embedding plane instead of the bumpy embedding surface which does not intersect atoms. A typical situation is shown in figure 2.9. The embedding surface S touches the atoms in our model. If we are interested in the Hamiltonian to the left of S, we shall have contributions from the muffin-tins to the left of S (see chapter 6), but we will have to take care of those parts of the Hamiltonian which belong to the space enclosed by

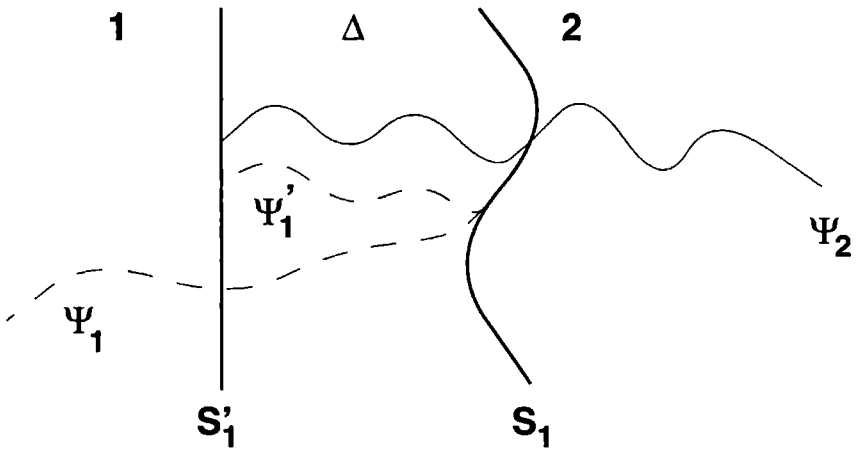


Figure 2.7: *In region Δ we have two potentials. V_1 is the potential of region 1, which is valid in Δ because Δ is part of region 1. V_2 is the potential of region 2 extended into Δ. The trial function Ψ_2 exists in region 2+Δ and the trial function Ψ_1 exists in region 1 and has two functional forms in region Δ, Ψ_1 and Ψ_1'.*

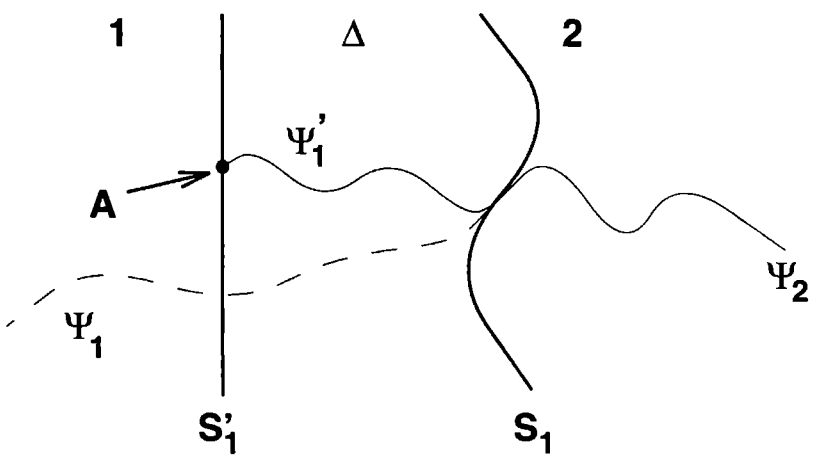


Figure 2.8: *We have imposed matching of Ψ_2 and Ψ_1' at A. This means that Ψ_2 and Ψ_1' have the same functional form in Δ, because they feel the same potential and thus are solutions of the same Schrodinger equation.*

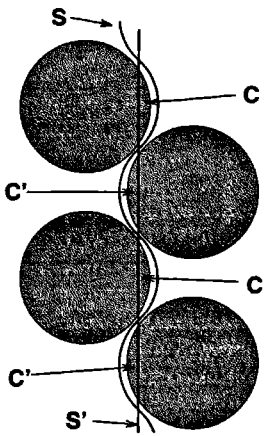


Figure 2.9: *S is the bumpy embedding surface that does not intersect any atoms. S' is the embedding plane that will be a lot easier to implement. C and C' indicates the caps which are the region Δ of this example.*

the muffin-tins on the left of S and the new surface S'. The caps indicated by C in figure 2.9 have to be subtracted (because we integrate *back* in those) and the caps indicated by C' have to be added (because we integrate *forward* in those).

We now have all the basic embedding knowledge we will need to apply the embedding approach to layered systems. This is the subject of the next chapter.

Chapter 3

Embedding layered systems

In this chapter we apply the embedding approach to layered systems. These have two-dimensional periodicity within the layers, which we can use. Also the embedding surfaces are all planes. We shall see that working with virtually independent layers, coupled by embedding potentials, enables us to create a method which scales linearly with the number of layers. We shall use embedding theory to find scattering properties, and conversely, we shall use scattering theory to find embedding potentials. An important first step is defining the geometry of the layered systems we want to handle.

3.1 Defining the geometry

In this section we discuss some of the reasons for working with layered systems. Also I shall point out some of the advantages of doing this, the most important being the improved scaling of the method.

3.1.1 Why use layers?

The use of layers in a calculation is almost automatically made when one wants to perform calculations for surfaces or interfaces, where the symmetry is broken in only one dimension. It is then useful to use layers parallel to the surface or interface.

3.1.2 What is a *subvolume*?

We have coined the word *subvolume*[6] to mean either a single layer or a grouping of layers. The subvolumes will be the units out of which we shall build our systems. Each subvolume will have its own basis, and the subvolumes will be coupled via embedding potentials.

3.1.3 Using the symmetry within the subvolume

Within the subvolume, we retain two-dimensional symmetry. The translational symmetry enables us to use a two-dimensional variation of Bloch's theorem:

$$\psi_{\mathbf{k}_\parallel}(\mathbf{r} + \mathbf{t}_\parallel) = e^{i\mathbf{k}_\parallel \cdot \mathbf{t}_\parallel} \psi_{\mathbf{k}_\parallel}(\mathbf{r}) \tag{3.1}$$

We will subsequently have to integrate over $\mathbf{k}_\parallel$ to find properties such as the charge density or total density of states. When we are interested in spectra measured using experimental methods like photoemission, we can use the density of states at certain $\mathbf{k}_\parallel$ to interpret these.

3.1.4 Why use the subvolumes with different bases?

Since we will be coupling the subvolumes through embedding potentials, this gives us the freedom to choose a different basis for each subvolume. The advantage of this is seen when we examine how the method scales with the number of subvolumes or layers.

We know that the size of a basis is more or less determined by the number and sort of atoms it has to handle. Suppose we choose N layers, all with the same one atom per layer, and suppose we need M basis functions per atom or layer. With a standard method, we would need $N \times M$ basis functions for the entire system. Solving the Schrödinger equation would then scale as $(N \times M)^3$, independent of whether we invert the Hamiltonian to get a Green function or diagonalise it to get eigenvalues/eigenfunctions. Using our embedding method with subvolumes, each layer/subvolume would scale as M^3, and we would need to do this N times. So embedding the subvolumes scales as $N \times M^3$. This is a very important result. Whereas standard methods scale with the cube of the number of layers in the calculation, the embedding method scales **linearly** with the number of layers. The price we have to pay for this is that we loose the off-diagonal elements of the Green function matrix between subvolumes, but this is information which is rarely used, certainly not in our applications (and in any case these elements can be retrieved using Green function techniques).

The basis set within each subvolume can be chosen according to the number and type of atoms within it. The only restriction in choosing the different basis sets in the different subvolumes is that the projections on the embedding planes of all bases have to match, to be able to couple the subvolumes through embedding potentials.

So now we have chosen a geometry, and we want to use subvolumes with the possibility of choosing a different basis for each. The next step is to explain in detail how embedding enables us to couple the subvolumes into one system.

3.2 Stacking subvolumes

Suppose we are trying to calculate a system which looks like the one sketched in figure 3.1. We have N subvolumes, embedded on the left and right by embedding potentials on S_L and S_R respectively. When we want to calculate the Green function in one of the subvolumes, we are going to need an embedding potential on both its left and right embedding surfaces.

3.2.1 Propagating embedding potentials

We want to propagate the left embedding potential on S_L to the left of every other subvolume. We do this in the following way:

- set the right embedding potential on S_1 to zero
- solve for the Green function in subvolume 1 by solving (see (2.25), but now we have two embedding planes S_L and S_R instead of one closed embedding surface S_1):

$$G_1 = \left[H_1 + \Sigma^L_{S_L} + \frac{1}{2}\frac{\partial}{\partial n_S}\bigg|_{S_L} + \frac{1}{2}\frac{\partial}{\partial n_S}\bigg|_{S_1} - EO_1 \right]^{-1} \tag{3.2}$$

where Σ^L indicates a **left** embedding potential, as Σ^R will indicate a **right** embedding potential. H_1 is the Hamiltonian in subvolume 1. The n_S normal on S_L points to the left of figure 3.1, the normal on S_1 points to the right.

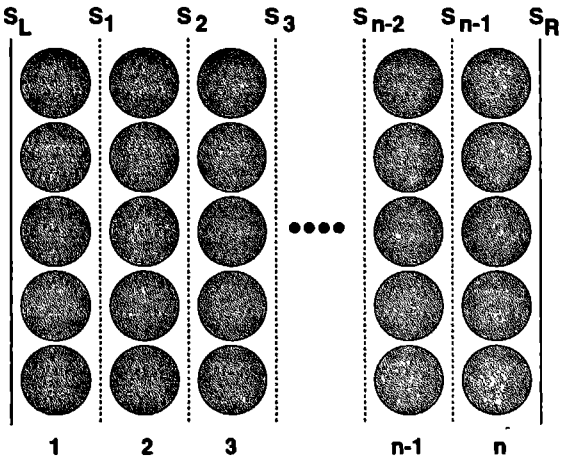


Figure 3.1: *Sketch of a geometry divided into n subvolumes.*

- G_1 will then have zero normal derivative on S_1, because we have set the embedding potential to zero on that plane.
- projecting G_1 onto a planar basis and surface inverting gives us $\Sigma^L_{S_1}$, the left-hand embedding potential for subvolume 2.

$$\Sigma^L_{S_1} = \left[C^R_{S_1} G_1 C^{R\dagger}_{S_1}\right]^{-1} \tag{3.3}$$

  Here $C^R_{S_1}$ is a basis projector, projecting the three-dimensional basis functions in volume 1 onto two-dimensional basis functions on surface S_1, to the right of 1. (3.3) is in effect the surface-inverse discussed in (2.6) and (2.5).

- set the right embedding potential on S_2 to zero
- solve for the Green function in subvolume 2 by solving (see (2.25) and (3.2)):

$$G_2 = \left[H_2 + \Sigma^L_{S_1} + \frac{1}{2}\frac{\partial}{\partial n_S}\bigg|_{S_1} + \frac{1}{2}\frac{\partial}{\partial n_S}\bigg|_{S_2} - EO_2\right]^{-1} \tag{3.4}$$

- G_2 will have zero normal derivative on S_2, because we have set the embedding potential to zero on that plane.
- projecting G_2 onto a planar basis and surface inverting gives us $\Sigma^L_{S_2}$, the left-hand embedding potential for subvolume 3, on embedding plane S_2.
- repeat this process until we have constructed all left-hand embedding potentials $\Sigma^L_{S_1}$ to $\Sigma^L_{S_{n-1}}$.

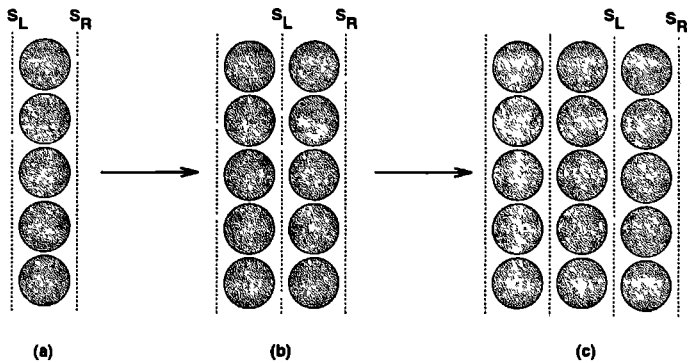


Figure 3.2: *Schematic picture of the iterative process for stacking subvolumes into a semi-infinite stack. New layers are added on the right and the positions of the embedding potentials for those layers are indicated.*

Similarly, we can propagate the right-hand embedding potential on S_R to the right of every subvolume. Once we have all the necessary embedding potentials, we can finally determine the Green function in each subvolume by solving:

$$G_i = \left[H_i + \Sigma^L_{S_{i-1}} + \frac{1}{2}\frac{\partial}{\partial n_S}\bigg|_{S_{i-1}} + \Sigma^R_{S_i} + \frac{1}{2}\frac{\partial}{\partial n_S}\bigg|_{S_i} - EO_i \right]^{-1} \tag{3.5}$$

where i runs from 1 to n and $S_0 = S_L$ and $S_n = S_R$. An important point to note in this algorithm, is that in order to obtain the Green function - which in each subvolume couples on both sides to its neighbours - we have to solve for three different Green functions, each time with different boundary conditions.

The geometry in figure 3.1 is applicable for example to a surface geometry, where on S_L we would embed a semi-infinite bulk, on S_R a semi-infinite vacuum, with subvolumes 1 to m filled with atoms, and subvolumes $m+1$ to n 'filled' with vacuum. From this it becomes clear that we need an algorithm for finding the embedding potential for a semi-infinite substrate, in other words a semi-infinite stack of identical subvolumes.

3.3 Finding the embedding potential for semi-infinite substrates

The approach to stacking the identical subvolumes representing the bulk is similar to the propagation process described in section 3.2.1. The difference is that we only propagate in one direction, towards the boundary of the substrate.

We start with a single subvolume, as shown in figure 3.2a. On S_L in this picture we apply an arbitrary embedding potential, in practice zero. On S_R we apply a zero embedding potential, because we want to have a zero normal derivative boundary condition on the Green

function. We solve for this Green function G_1, just as we did before ((2.25), (3.2)):

$$G_1 = \left[H_1 + \frac{1}{2}\frac{\partial}{\partial n_S}\Big|_{S_L} + \frac{1}{2}\frac{\partial}{\partial n_S}\Big|_{S_R} - EO_1 \right]^{-1} \tag{3.6}$$

From this Green function we construct a left-hand embedding potential, which we call Σ_1^L:

$$\Sigma_1^L = \left[C_{S_1}^R G_1 C_{S_1}^{R\dagger} \right]^{-1} \tag{3.7}$$

This time the index of the left embedding potential tells us about the number of subvolumes to the left of it.

We then add a second subvolume, and arrive at figure 3.2b. We have moved the embedding planes S_L and S_R to the second subvolume. On S_L we apply the embedding potential which we have just found, Σ_1^L. On the right of the second subvolume we again apply the zero embedding potential, to get zero normal derivative in the resulting Green function. This Green function we call G_2.

$$G_2 = \left[H_2 + \Sigma_1^L + \frac{1}{2}\frac{\partial}{\partial n_S}\Big|_{S_L} + \frac{1}{2}\frac{\partial}{\partial n_S}\Big|_{S_R} - EO_2 \right]^{-1} \tag{3.8}$$

We can again get an embedding potential from this Green function, Σ_2^L. This is now an embedding potential of 2 subvolumes.

We again add a subvolume and get figure 3.2c. As before, on S_L we apply the embedding potential found previously Σ_2^L, and on S_R a zero embedding potential. The Green function for the third subvolume,

$$G_3 = \left[H_3 + \Sigma_2^L + \frac{1}{2}\frac{\partial}{\partial n_S}\Big|_{S_L} + \frac{1}{2}\frac{\partial}{\partial n_S}\Big|_{S_R} - EO_3 \right]^{-1}, \tag{3.9}$$

again gives us an embedding potential, namely Σ_3^L.

We now repeat this iterative procedure, each iteration giving us the embedding potential for a stack that is one subvolume thicker than before. The effects of the boundary at S_L in figure 3.2a become less and less, and eventually the process converges to give the solution for the semi-infinite stack. We will call it converged when the difference between the representations of successive embedding potentials Σ_N^L and Σ_{N+1}^L is smaller than some given tolerance. In practice the number of subvolumes we need to stack before reaching convergence will be of the order of hundreds. This means hundreds of steps, each one involving an inversion to find a Green function. In other words, this approach is not very efficient. The logical next step is to try and reach convergence with fewer inversions.

3.3.1 How to speed up convergence

What we would like to have is a closed expression for the $(N+1)^{th}$ embedding potential as some multi-dimensional function F of the N^{th}:

$$\Sigma_{N+1}^L = F(\Sigma_N^L) \tag{3.10}$$

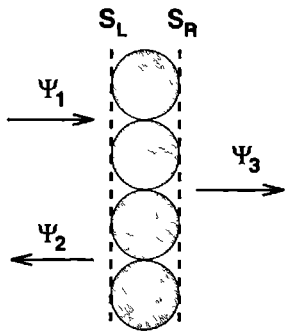


Figure 3.3: *A single subvolume with embedding planes and an incoming wave Ψ_1 from the left being scattered into outgoing waves Ψ_2 and Ψ_3.*

The problem of finding the solution for the semi-infinite stack is then one of finding the root of the multi-dimensional function:

$$F(\Sigma_N^L) - \Sigma_N^L = 0 \tag{3.11}$$

because we know that Σ_{N+1}^L equals Σ_N^L when we have reached convergence. Newton-Raphson-like approaches have been developed for this type of problem[5, 7], but even with such an acceleration scheme, it still takes about 5 iterations per energy and per k-point to find the embedding potential. On implementation, this turned out to be one of the more time consuming parts of the calculation, so other ways had to be found to find the embedding potential.

3.4 Embedding and scattering theory

The major problem with all processes involving Green functions in any implementation of the embedding method is that the Green functions are defined in the full three-dimensional basis. This basis is typically of the order of a hundred functions per atom. Thus the inversion of the Hamiltonian to find the Green function is a time consuming process, especially if it has to be done several times per energy and k-point. Now, the embedding potential is defined in terms of a much smaller two-dimensional basis on the embedding planes. So the question arises, can we somehow use quantities that are also defined in terms of this two-dimensional basis to find the embedding potential. An obvious choice are the reflection and transmission properties of the subvolumes. In the next section we will derive the reflection and transmission matrices from the Green function for a subvolume.

3.4.1 From embedding to scattering

Let us assume that we have a single subvolume as shown in figure 3.3. We have an incoming wave, Ψ_1, and two outgoing waves, one reflected, Ψ_2, and one transmitted, Ψ_3.
The incoming wave Ψ_1 we write as:

$$\Psi_1(\mathbf{r}) = e^{\gamma_{\mathbf{G}}^{+} z} e^{i(\mathbf{k}_\parallel + \mathbf{G})\,\mathbf{R}} \qquad \text{for certain } \mathbf{k}_\parallel \text{ and } \mathbf{G} \tag{3.12}$$

where

$$\gamma_{\mathbf{G}}^{\pm} = \mp\sqrt{|\mathbf{k}_{\parallel} + \mathbf{G}|^2 - 2E} \tag{3.13}$$

with E the energy. The $\pm$ sign on γ depends on the direction in which the wave is travelling; in the case of Ψ_1 this is a + sign. We have to choose the branch of the complex square root such that for γ^+ the imaginary part is positive and for γ^- the imaginary part is negative. $\mathbf{k}_{\parallel}$ is the k-vector which we have to incorporate because we are using two-dimensional Bloch's theorem in the subvolume. $\mathbf{G}$ is a vector of the two-dimensional basis defined on the planes S_R and S_L, and is a reciprocal lattice vector of the two-dimensional lattice of the subvolume. The reflected wave Ψ_2 we write as:

$$\Psi_2(\mathbf{r}) = \sum_{\mathbf{G}'} R^L_{\mathbf{G}'\mathbf{G}} e^{\gamma^-_{\mathbf{G}'} z} e^{\imath(\mathbf{k}_{\parallel}+\mathbf{G}')\,\mathbf{R}} \tag{3.14}$$

and the transmitted wave as:

$$\Psi_3(\mathbf{r}) = \sum_{\mathbf{G}'} T^L_{\mathbf{G}'\mathbf{G}} e^{\gamma^+_{\mathbf{G}'} z} e^{\imath(\mathbf{k}_{\parallel}+\mathbf{G}')\,\mathbf{R}} \tag{3.15}$$

Here we take the z-origins of the waves on the left and the right of the subvolume to be on the left and right respectively. In other words, for a wave on the left of the subvolume, the z-origin will be on the left-hand embedding plane of the subvolume. Because of this choice of z-origins, the expressions for the waves on the embedding planes will not contain an exponent of z. In a first derivation the z origin was chosen to be in the origin of the subvolume, but for thick subvolumes this led to instabilities because of the enormous ratio between exponentially increasing and decreasing waves.
We shall now define a Ψ_{left} and a Ψ_{right} to see how these relate to the subvolume Green function.

$$\begin{aligned}\Psi_{\text{left}}(\mathbf{r}) &= \Psi_1(\mathbf{r}) + \Psi_2(\mathbf{r}) \\ &= e^{\imath\mathbf{k}_{\parallel}\,\mathbf{R}} \times \left[e^{\gamma^+_{\mathbf{G}} z} e^{\imath\mathbf{G}\,\mathbf{R}} + \sum_{\mathbf{G}'} R^L_{\mathbf{G}'\mathbf{G}} e^{\gamma^-_{\mathbf{G}'} z} e^{\imath\mathbf{G}'\,\mathbf{R}} \right] \end{aligned} \tag{3.16a}$$

$$\begin{aligned}\Psi_{\text{right}}(\mathbf{r}) &= \Psi_3(\mathbf{r}) \\ &= e^{\imath\mathbf{k}_{\parallel}\,\mathbf{R}} \times \left[\sum_{\mathbf{G}'} T^L_{\mathbf{G}'\mathbf{G}} e^{\gamma^+_{\mathbf{G}'} z} e^{\imath\mathbf{G}'\,\mathbf{R}} \right] \end{aligned} \tag{3.16b}$$

Since we want to relate these to the Green function through a logarithmic derivative, we also need the normal derivatives of both (the normal derivative is defined with a direction into the subvolume):

$$\begin{aligned}\frac{\partial}{\partial n_S}\Psi_{\text{left}}(\mathbf{r}) &= \frac{\mathrm{d}}{\mathrm{d}z}\Psi_{\text{left}}(\mathbf{r}) \\ &= e^{\imath\mathbf{k}_{\parallel}\,\mathbf{R}} \times \left[\gamma^+_{\mathbf{G}} e^{\gamma^+_{\mathbf{G}} z} e^{\imath\mathbf{G}\,\mathbf{R}} + \sum_{\mathbf{G}'} R^L_{\mathbf{G}'\mathbf{G}} \gamma^-_{\mathbf{G}'} e^{\gamma^-_{\mathbf{G}'} z} e^{\imath\mathbf{G}'\,\mathbf{R}} \right] \end{aligned} \tag{3.17a}$$

and:

$$\frac{\partial}{\partial n_S}\Psi_{\text{right}}(\mathbf{r}) = -\frac{\mathrm{d}}{\mathrm{d}z}\Psi_{\text{right}}(\mathbf{r})$$
$$= -e^{i\mathbf{k}_{\|}\cdot\mathbf{R}} \times \left[\sum_{\mathbf{G}'} T^{L}_{\mathbf{G}'\mathbf{G}}\gamma^{+}_{\mathbf{G}'} e^{\gamma^{+}_{\mathbf{G}'}z} e^{i\mathbf{G}'\cdot\mathbf{R}}\right] \tag{3.17b}$$

We would like to have a relationship between the Ψ's and their derivatives through the Green function of the subvolume. To obtain this relationship, we write down the following expressions of the Schrödinger equation for the subvolume:

$$\left(-\tfrac{1}{2}\nabla^2 + V(\mathbf{r}) - E\right) G(\mathbf{r}', \mathbf{r}) = \delta(\mathbf{r} - \mathbf{r}') \tag{3.18a}$$
$$\left(-\tfrac{1}{2}\nabla^2 + V(\mathbf{r}) - E\right) \Psi(\mathbf{r}) = 0 \tag{3.18b}$$

We multiply (3.18a) from the left with $\Psi(\mathbf{r})$ and multiply (3.18b) from the left with $G(\mathbf{r}', \mathbf{r})$, to obtain:

$$\Psi(\mathbf{r})\left(-\tfrac{1}{2}\nabla^2 + V(\mathbf{r}) - E\right) G(\mathbf{r}', \mathbf{r}) = \Psi(\mathbf{r})\delta(\mathbf{r} - \mathbf{r}') \tag{3.19a}$$
$$G(\mathbf{r}', \mathbf{r})\left(-\tfrac{1}{2}\nabla^2 + V(\mathbf{r}) - E\right) \Psi(\mathbf{r}) = 0 \tag{3.19b}$$

Subtracting (3.19b) from (3.19a) and integrating over $\mathbf{r}$ gives us:

$$-\tfrac{1}{2}\int \mathrm{d}^3\mathbf{r}\left(\Psi(\mathbf{r})\nabla^2 G(\mathbf{r}', \mathbf{r}) - G(\mathbf{r}', \mathbf{r})\nabla^2\Psi(\mathbf{r})\right) = \int \mathrm{d}^3\mathbf{r}\Psi(\mathbf{r})\delta(\mathbf{r} - \mathbf{r}') \tag{3.20}$$

Using Green's theorem, we can write this as:

$$-\tfrac{1}{2}\left(\int_{S_L} \mathrm{d}^2\mathbf{r}_S G(\mathbf{r}', \mathbf{r}_S)\frac{\partial}{\partial n_S}\Psi(\mathbf{r}_S) + \int_{S_R} \mathrm{d}^2\mathbf{r}_S G(\mathbf{r}', \mathbf{r}_S)\frac{\partial}{\partial n_S}\Psi(\mathbf{r}_S)\right) = \Psi(\mathbf{r}') \tag{3.21}$$

where we have used the fact that we impose the following boundary conditions on the Green function:

$$\frac{\partial}{\partial n_S}G(\mathbf{r}, \mathbf{r}_S)\Big|_{S_L} = 0 \qquad \frac{\partial}{\partial n_S}G(\mathbf{r}, \mathbf{r}_S)\Big|_{S_R} = 0 \tag{3.22}$$

We can now choose $\mathbf{r}'$ to be on the left, $\mathbf{r}' = \mathbf{r}_L$ or on the right $\mathbf{r}' = \mathbf{r}_R$, to obtain:

$$\Psi(\mathbf{r}_L) = -\tfrac{1}{2}\left[\int_{S_L} \mathrm{d}^2\mathbf{r}_S G(\mathbf{r}_L, \mathbf{r}_S)\frac{\partial}{\partial n_S}\Psi(\mathbf{r}_S) + \int_{S_R} \mathrm{d}^2\mathbf{r}_S G(\mathbf{r}_L, \mathbf{r}_S)\frac{\partial}{\partial n_S}\Psi(\mathbf{r}_S)\right] \tag{3.23a}$$

$$\Psi(\mathbf{r}_R) = -\tfrac{1}{2}\left[\int_{S_L} \mathrm{d}^2\mathbf{r}_S G(\mathbf{r}_R, \mathbf{r}_S)\frac{\partial}{\partial n_S}\Psi(\mathbf{r}_S) + \int_{S_R} \mathrm{d}^2\mathbf{r}_S G(\mathbf{r}_R, \mathbf{r}_S)\frac{\partial}{\partial n_S}\Psi(\mathbf{r}_S)\right] \tag{3.23b}$$

where $\Psi(\mathbf{r})$ is the wavefunction just inside the subvolume, matching in logarithmic derivative to $\Psi_{\text{left}}(\mathbf{r})$ on S_L and matching in logarithmic derivative to $\Psi_{\text{right}}(\mathbf{r})$ on S_R. Substituting on these surfaces, we obtain:

$$\Psi_{\text{left}}(\mathbf{r}_L) = -\tfrac{1}{2}\left[\int_{S_L} \mathrm{d}^2\mathbf{r}_S G(\mathbf{r}_L, \mathbf{r}_S)\frac{\partial}{\partial n_S}\Psi_{\text{left}}(\mathbf{r}_S)\right.$$
$$\left. + \int_{S_R} \mathrm{d}^2\mathbf{r}_S G(\mathbf{r}_L, \mathbf{r}_S)\frac{\partial}{\partial n_S}\Psi_{\text{right}}(\mathbf{r}_S)\right] \tag{3.24a}$$

$$\Psi_{\text{right}}(\mathbf{r}_R) = -\frac{1}{2}\left[\int_{S_L} \mathrm{d}^2\mathbf{r}_S G(\mathbf{r}_R,\mathbf{r}_S)\frac{\partial}{\partial n_S}\Psi_{\text{left}}(\mathbf{r}_S) + \int_{S_R} \mathrm{d}^2\mathbf{r}_S G(\mathbf{r}_R,\mathbf{r}_S)\frac{\partial}{\partial n_S}\Psi_{\text{right}}(\mathbf{r}_S)\right] \quad (3.24b)$$

The Green function in these expressions can be expanded in terms of basis functions within the subvolume:

$$G(\mathbf{r},\mathbf{r}') = \sum_{\mathbf{g}\mathbf{g}'} \chi_{\mathbf{g}}(\mathbf{r}) G_{\mathbf{g}\mathbf{g}'} \chi^*_{\mathbf{g}'}(\mathbf{r}') \quad (3.25)$$

Writing out (3.24a), using (3.17a) , (3.17b), and the expansion of (3.25) we obtain:

$$\begin{aligned} e^{\imath\mathbf{k}_\parallel \mathbf{R}_L}\left[e^{\imath\mathbf{G}\,\mathbf{R}_L} + \sum_{\mathbf{G}'} R^L_{\mathbf{G}'\mathbf{G}} e^{\imath\mathbf{G}'\,\mathbf{R}_L}\right] = \\ -\frac{1}{2}\sum_{\mathbf{g}\mathbf{g}'}\chi_{\mathbf{g}}(\mathbf{r}_L)G_{\mathbf{g}\mathbf{g}'}\gamma^+_{\mathbf{G}}\int_{S_L}\mathrm{d}^2\mathbf{r}_S\chi^*_{\mathbf{g}'}(\mathbf{r}_S)e^{\imath\mathbf{k}_\parallel\mathbf{R}_S}e^{\imath\mathbf{G}\,\mathbf{R}_S} \\ -\frac{1}{2}\sum_{\mathbf{g}\mathbf{g}'}\sum_{\mathbf{G}'}\chi_{\mathbf{g}}(\mathbf{r}_L)G_{\mathbf{g}\mathbf{g}'}R^L_{\mathbf{G}'\mathbf{G}}\gamma^-_{\mathbf{G}'}\int_{S_L}\mathrm{d}^2\mathbf{r}_S\chi^*_{\mathbf{g}'}(\mathbf{r}_S)e^{\imath\mathbf{k}_\parallel\mathbf{R}_S}e^{\imath\mathbf{G}'\,\mathbf{R}_S} \\ +\frac{1}{2}\sum_{\mathbf{g}\mathbf{g}'}\sum_{\mathbf{G}'}\chi_{\mathbf{g}}(\mathbf{r}_L)G_{\mathbf{g}\mathbf{g}'}T^L_{\mathbf{G}'\mathbf{G}}\gamma^+_{\mathbf{G}'}\int_{S_R}\mathrm{d}^2\mathbf{r}_S\chi^*_{\mathbf{g}'}(\mathbf{r}_S)e^{\imath\mathbf{k}_\parallel\mathbf{R}_S}e^{\imath\mathbf{G}'\,\mathbf{R}_S} \quad (3.26) \end{aligned}$$

Multiplying (3.26) with $e^{-\imath\mathbf{k}_\parallel\mathbf{R}_L}e^{-\imath\mathbf{G}_1\mathbf{R}_L}$ and integrating $\mathbf{R}_L$ over the surface S_L, we get:

$$\begin{aligned} \int_{S_L}\mathrm{d}^2\mathbf{R}_L e^{-\imath\mathbf{G}_1\mathbf{R}_L}e^{\imath\mathbf{G}\,\mathbf{R}_L} + \sum_{\mathbf{G}'}R^L_{\mathbf{G}'\mathbf{G}}\int_{S_L}\mathrm{d}^2\mathbf{R}_L e^{-\imath\mathbf{G}_1\mathbf{R}_L}e^{\imath\mathbf{G}'\,\mathbf{R}_L} = \\ -\frac{1}{2}\sum_{\mathbf{g}\mathbf{g}'}\int_{S_L}\mathrm{d}^2\mathbf{R}_L e^{-\imath\mathbf{k}_\parallel\mathbf{R}_L}e^{-\imath\mathbf{G}_1\mathbf{R}_L}\chi_{\mathbf{g}}(\mathbf{r}_L)G_{\mathbf{g}\mathbf{g}'}\gamma^+_{\mathbf{G}}\times \\ \int_{S_L}\mathrm{d}^2\mathbf{R}_S\chi^*_{\mathbf{g}'}(\mathbf{R}_S)e^{\imath\mathbf{k}_\parallel\mathbf{R}_S}e^{\imath\mathbf{G}\,\mathbf{R}_S} \\ -\frac{1}{2}\sum_{\mathbf{g}\mathbf{g}'}\sum_{\mathbf{G}'}\int_{S_L}\mathrm{d}^2\mathbf{R}_L e^{-\imath\mathbf{k}_\parallel\mathbf{R}_L}e^{-\imath\mathbf{G}_1\mathbf{R}_L}\chi_{\mathbf{g}}(\mathbf{r}_L)G_{\mathbf{g}\mathbf{g}'}R^L_{\mathbf{G}'\mathbf{G}}\gamma^-_{\mathbf{G}'}\times \\ \int_{S_L}\mathrm{d}^2\mathbf{R}_S\chi^*_{\mathbf{g}'}(\mathbf{R}_S)e^{\imath\mathbf{k}_\parallel\mathbf{R}_S}e^{\imath\mathbf{G}'\,\mathbf{R}_S} \\ +\frac{1}{2}\sum_{\mathbf{g}\mathbf{g}'}\sum_{\mathbf{G}'}\int_{S_L}\mathrm{d}^2\mathbf{R}_L e^{-\imath\mathbf{k}_\parallel\mathbf{R}_L}e^{-\imath\mathbf{G}_1\mathbf{R}_L}\chi_{\mathbf{g}}(\mathbf{r}_L)G_{\mathbf{g}\mathbf{g}'}T^L_{\mathbf{G}'\mathbf{G}}\gamma^+_{\mathbf{G}'}\times \\ \int_{S_R}\mathrm{d}^2\mathbf{R}_S\chi^*_{\mathbf{g}'}(\mathbf{R}_S)e^{\imath\mathbf{k}_\parallel\mathbf{R}_S}e^{\imath\mathbf{G}'\,\mathbf{R}_S} \quad (3.27) \end{aligned}$$

This can be written as a matrix equation, by defining the following matrices:

$$\mathbb{M}^{L/R}_{\mathbf{G}_1\mathbf{G}} = \frac{1}{A}\int_{S_{L/R}}\mathrm{d}^2\mathbf{R}_S e^{\imath(\mathbf{G}-\mathbf{G}_1)\,\mathbf{R}_S} = \delta_{\mathbf{G}_1\mathbf{G}} \quad (3.28a)$$

$$\mathbb{N}^{L/R}_{\mathbf{G}_1\mathbf{G}} = \frac{1}{A}\int_{S_{L/R}}\mathrm{d}^2\mathbf{R}_S e^{\imath(\mathbf{G}-\mathbf{G}_1)\,\mathbf{R}_S} = \delta_{\mathbf{G}_1\mathbf{G}} \quad (3.28b)$$

$$\mathbb{C}^{L/R}_{\mathbf{G}_1\mathbf{g}} = \frac{1}{\sqrt{A}}\int_{S_{L/R}}\mathrm{d}^2\mathbf{R}_S e^{-\imath(\mathbf{k}_\parallel+\mathbf{G}_1)\,\mathbf{R}_S}\chi_{\mathbf{g}}(\mathbf{R}_S) \quad (3.28c)$$

and vectors:

$$\mathbb{V}^L_{\mathbf{G}} = \gamma^+_{\mathbf{G}} \tag{3.28d}$$

$$\mathbb{W}^L_{\mathbf{G}} = \gamma^-_{\mathbf{G}} \tag{3.28e}$$

$$\mathbb{V}^R_{\mathbf{G}} = \gamma^+_{\mathbf{G}} \tag{3.28f}$$

$$\mathbb{W}^R_{\mathbf{G}} = \gamma^-_{\mathbf{G}} \tag{3.28g}$$

Using these definitions, (3.27) becomes:

$$\begin{aligned}\mathbb{M}^L_{\mathbf{G}_1\mathbf{G}} + \sum_{\mathbf{G}'} \mathbb{N}^L_{\mathbf{G}_1\mathbf{G}'} R^L_{\mathbf{G}'\mathbf{G}} = &-\tfrac{1}{2}\sum_{\mathbf{g}\mathbf{g}'} C^L_{\mathbf{G}_1\mathbf{g}} G_{\mathbf{g}\mathbf{g}'} C^{L\dagger}_{\mathbf{g}'\mathbf{G}} \mathbb{V}^L_{\mathbf{G}} \\ &-\tfrac{1}{2}\sum_{\mathbf{g}\mathbf{g}'}\sum_{\mathbf{G}'} C^L_{\mathbf{G}_1\mathbf{g}} G_{\mathbf{g}\mathbf{g}'} C^{L\dagger}_{\mathbf{g}'\mathbf{G}'} \mathbb{W}^L_{\mathbf{G}'} R^L_{\mathbf{G}'\mathbf{G}} \\ &+\tfrac{1}{2}\sum_{\mathbf{g}\mathbf{g}'}\sum_{\mathbf{G}'} C^L_{\mathbf{G}_1\mathbf{g}} G_{\mathbf{g}\mathbf{g}'} C^{R\dagger}_{\mathbf{g}'\mathbf{G}'} \mathbb{V}^R_{\mathbf{G}'} T^L_{\mathbf{G}'\mathbf{G}}\end{aligned} \tag{3.29}$$

which is the following matrix equation:

$$\mathbb{M}^L + \mathbb{N}^L R^L = -\tfrac{1}{2} C^L G C^{L\dagger} \mathbb{V}^L - \tfrac{1}{2} C^L G C^{L\dagger} \mathbb{W}^L R^L + \tfrac{1}{2} C^L G C^{R\dagger} \mathbb{V}^R T^L \tag{3.30}$$

Writing out equation (3.24b) gives:

$$\begin{aligned}e^{\imath \mathbf{k}_\parallel \mathbf{R}_R} \sum_{\mathbf{G}'} T^L_{\mathbf{G}'\mathbf{G}} e^{\imath \mathbf{G}' \mathbf{R}_R} = \\ -\tfrac{1}{2}\sum_{\mathbf{g}\mathbf{g}'} \chi_{\mathbf{g}}(\mathbf{r}_R) G_{\mathbf{g}\mathbf{g}'} \gamma^+_{\mathbf{G}} \int_{S_L} \mathrm{d}^2\mathbf{R}_S \chi^*_{\mathbf{g}'}(\mathbf{R}_S) e^{\imath \mathbf{k}_\parallel \mathbf{R}_S} e^{\imath \mathbf{G} \mathbf{R}_S} \\ -\tfrac{1}{2}\sum_{\mathbf{g}\mathbf{g}'}\sum_{\mathbf{G}'} \chi_{\mathbf{g}}(\mathbf{r}_R) G_{\mathbf{g}\mathbf{g}'} R^L_{\mathbf{G}'\mathbf{G}} \gamma^-_{\mathbf{G}'} \int_{S_L} \mathrm{d}^2\mathbf{R}_S \chi^*_{\mathbf{g}'}(\mathbf{R}_S) e^{\imath \mathbf{k}_\parallel \mathbf{R}_S} e^{\imath \mathbf{G}' \mathbf{R}_S} \\ +\tfrac{1}{2}\sum_{\mathbf{g}\mathbf{g}'}\sum_{\mathbf{G}'} \chi_{\mathbf{g}}(\mathbf{r}_R) G_{\mathbf{g}\mathbf{g}'} T^L_{\mathbf{G}'\mathbf{G}} \gamma^+_{\mathbf{G}'} \int_{S_R} \mathrm{d}^2\mathbf{R}_S \chi^*_{\mathbf{g}'}(\mathbf{R}_S) e^{\imath \mathbf{k}_\parallel \mathbf{R}_S} e^{\imath \mathbf{G}' \mathbf{R}_S}\end{aligned} \tag{3.31}$$

multiplying (3.31) by $e^{-\imath \mathbf{k}_\parallel \mathbf{R}_R} e^{-\imath \mathbf{G}_1 \mathbf{R}_R}$ and integrating $\mathbf{R}_R$ over S_R, we get:

$$\begin{aligned}\sum_{\mathbf{G}'} T^L_{\mathbf{G}'\mathbf{G}} \int_{S_R} \mathrm{d}^2\mathbf{R}_R \, e^{-\imath \mathbf{G}_1 \mathbf{R}_R} \, e^{\imath \mathbf{G}' \mathbf{R}_R} = \\ -\tfrac{1}{2}\sum_{\mathbf{g}\mathbf{g}'} \int_{S_R} \mathrm{d}^2\mathbf{R}_R e^{-\imath \mathbf{k}_\parallel \mathbf{R}_R} e^{-\imath \mathbf{G}_1 \mathbf{R}_R} \chi_{\mathbf{g}}(\mathbf{r}_R) G_{\mathbf{g}\mathbf{g}'} \gamma^+_{\mathbf{G}} \times \\ \int_{S_L} \mathrm{d}^2\mathbf{R}_S \chi^*_{\mathbf{g}'}(\mathbf{R}_S) e^{\imath \mathbf{k}_\parallel \mathbf{R}_S} e^{\imath \mathbf{G} \mathbf{R}_S} \\ -\tfrac{1}{2}\sum_{\mathbf{g}\mathbf{g}'}\sum_{\mathbf{G}'} \int_{S_R} \mathrm{d}^2\mathbf{R}_R e^{-\imath \mathbf{k}_\parallel \mathbf{R}_R} e^{-\imath \mathbf{G}_1 \mathbf{R}_R} \chi_{\mathbf{g}}(\mathbf{r}_R) G_{\mathbf{g}\mathbf{g}'} R^L_{\mathbf{G}'\mathbf{G}} \gamma^-_{\mathbf{G}'} \times \\ \int_{S_L} \mathrm{d}^2\mathbf{R}_S \chi^*_{\mathbf{g}'}(\mathbf{R}_S) e^{\imath \mathbf{k}_\parallel \mathbf{R}_S} e^{\imath \mathbf{G}' \mathbf{R}_S} \\ +\tfrac{1}{2}\sum_{\mathbf{g}\mathbf{g}'}\sum_{\mathbf{G}'} \int_{S_R} \mathrm{d}^2\mathbf{R}_R e^{-\imath \mathbf{k}_\parallel \mathbf{R}_R} e^{-\imath \mathbf{G}_1 \mathbf{R}_R} \chi_{\mathbf{g}}(\mathbf{r}_R) G_{\mathbf{g}\mathbf{g}'} T^L_{\mathbf{G}'\mathbf{G}} \gamma^+_{\mathbf{G}'} \times \\ \int_{S_R} \mathrm{d}^2\mathbf{R}_S \chi^*_{\mathbf{g}'}(\mathbf{R}_S) e^{\imath \mathbf{k}_\parallel \mathbf{R}_S} e^{\imath \mathbf{G}' \mathbf{R}_S}\end{aligned} \tag{3.32}$$

Using the definitions of matrices and vectors in (3 28a)-(3 28g) we write (3 32) as

$$\begin{aligned}\sum_{\mathbf{G}'}\mathbb{M}^R_{\mathbf{G}_1\mathbf{G}'}T^L_{\mathbf{G}'\mathbf{G}} = &-\tfrac{1}{2}\sum_{\mathbf{g}\mathbf{g}'}C^R_{\mathbf{G}_1\mathbf{g}}G_{\mathbf{g}\mathbf{g}'}C^{L\dagger}_{\mathbf{g}'\mathbf{G}}\mathbb{V}^L_{\mathbf{G}}\\ &-\tfrac{1}{2}\sum_{\mathbf{g}\mathbf{g}'}\sum_{\mathbf{G}'}C^R_{\mathbf{G}_1\mathbf{g}}G_{\mathbf{g}\mathbf{g}'}C^{L\dagger}_{\mathbf{g}'\mathbf{G}'}\mathbb{W}^L_{\mathbf{G}'}R^L_{\mathbf{G}'\mathbf{G}}\\ &+\tfrac{1}{2}\sum_{\mathbf{g}\mathbf{g}'}\sum_{\mathbf{G}'}C^R_{\mathbf{G}_1\mathbf{g}}G_{\mathbf{g}\mathbf{g}'}C^{R\dagger}_{\mathbf{g}'\mathbf{G}'}\mathbb{V}^R_{\mathbf{G}'}T^L_{\mathbf{G}'\mathbf{G}}\end{aligned} \tag{3 33}$$

As a matrix equation this becomes

$$\mathbb{M}^R T^L = -\tfrac{1}{2}C^R G C^{L\dagger}\mathbb{V}^L - \tfrac{1}{2}C^R G C^{L\dagger}\mathbb{W}^L R^L + \tfrac{1}{2}C^R G C^{R\dagger}\mathbb{V}^R T^L \tag{3 34}$$

We now define the following matrices

$$\mathbb{A}^L = -\tfrac{1}{2}C^L G C^{L\dagger}\mathbb{V}^L \tag{3 35a}$$
$$\mathbb{B}^L = -\tfrac{1}{2}C^L G C^{L\dagger}\mathbb{W}^L \tag{3 35b}$$
$$\mathbb{C}^L = +\tfrac{1}{2}C^L G C^{R\dagger}\mathbb{V}^R \tag{3 35c}$$
$$\mathbb{D}^L = -\tfrac{1}{2}C^R G C^{L\dagger}\mathbb{V}^L \tag{3 35d}$$
$$\mathbb{E}^L = -\tfrac{1}{2}C^R G C^{L\dagger}\mathbb{W}^L \tag{3 35e}$$
$$\mathbb{F}^L = +\tfrac{1}{2}C^R G C^{R\dagger}\mathbb{V}^R \tag{3 35f}$$

This leads to the following set of matrix equations, from (3 30) and (3 34)

$$\mathbb{M}^L + \mathbb{N}^L R^L = \mathbb{A}^L + \mathbb{B}^L R^L + \mathbb{C}^L T^L \tag{3 36a}$$
$$\mathbb{M}^R T^L = \mathbb{D}^L + \mathbb{E}^L R^L + \mathbb{F}^L T^L \tag{3 36b}$$

We can now express T^L in terms of R^L, using (3 36b)

$$T^L = \left(\mathbb{M}^R - \mathbb{F}^L\right)^{-1} \times \left(\mathbb{D}^L + \mathbb{E}^L R^L\right) \tag{3 37}$$

If we put (3 37) into (3 36a), we can solve for R^L

$$R^L = \left[\mathbb{N}^L - \mathbb{B}^L - \mathbb{C}^L\left(\mathbb{M}^R - \mathbb{F}^L\right)^{-1}\mathbb{E}^L\right]^{-1}\left(\mathbb{A}^L - \mathbb{M}^L + \mathbb{C}^L\left(\mathbb{M}^R - \mathbb{F}^L\right)^{-1}\mathbb{D}^L\right) \tag{3 38}$$

Once we have calculated R^L, we use (3 37) to get T^L This matrix equation has the dimensions of the two-dimensional basis, which means that it is rather quick to determine

We have thus used the embedding approach, or actually just Green's theorem, to derive expressions for R^L and T^L in terms of the subvolume Green function G To derive similar expression for R^R and T^R, we have to look at the situation in figure 3 4, where the direction of the incoming wave has been changed

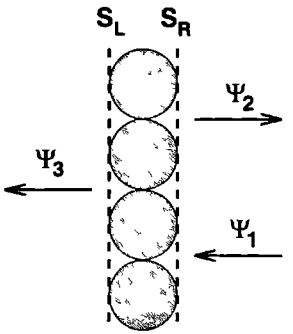


Figure 3.4: *A single subvolume with embedding planes and an incoming wave Ψ_1 from the right being scattered into outgoing waves Ψ_2 and Ψ_3.*

The incoming wave is now:

$$\Psi_1(\mathbf{r}) = e^{\gamma_{\mathbf{G}}^- z} e^{\imath(\mathbf{k}_\parallel + \mathbf{G})\,\mathbf{R}} \tag{3.39}$$

The reflected wave is:

$$\Psi_2(\mathbf{r}) = \sum_{\mathbf{G}'} R^R_{\mathbf{G}'\mathbf{G}} e^{\gamma_{\mathbf{G}'}^+ z} e^{\imath(\mathbf{k}_\parallel + \mathbf{G}')\,\mathbf{R}} \tag{3.40}$$

and the transmitted wave:

$$\Psi_3(\mathbf{r}) = \sum_{\mathbf{G}'} T^R_{\mathbf{G}'\mathbf{G}} e^{\gamma_{\mathbf{G}'}^- z} e^{\imath(\mathbf{k}_\parallel + \mathbf{G}')\,\mathbf{R}} \tag{3.41}$$

As in the case of the incoming wave from the left we obtain the following matrix equations:

$$\mathbb{N}^L T^R = \mathbb{A}^R + \mathbb{B}^R R^R + \mathbb{C}^R T^R \tag{3.42a}$$

$$\mathbb{N}^R + \mathbb{M}^R R^R = \mathbb{D}^R + \mathbb{E}^R R^R + \mathbb{F}^R T^R \tag{3.42b}$$

We can express T^R in terms of R^R from (3.42a).

$$T^R = \left(\mathbb{N}^L - \mathbb{C}^R\right)^{-1} \times \left(\mathbb{A}^R + \mathbb{B}^R R^R\right) \tag{3.43}$$

and for R^R we now get the following equation:

$$R^R = \left[\mathbb{M}^R - \mathbb{E}^R - \mathbb{F}^R \left(\mathbb{N}^L - \mathbb{C}^R\right)^{-1} \mathbb{B}^R\right]^{-1} \left(\mathbb{D}^R - \mathbb{N}^R + \mathbb{F}^R \left(\mathbb{N}^L - \mathbb{C}^R\right)^{-1} \mathbb{A}^R\right) \tag{3.44}$$

In this section we have shown how to construct scattering matrices for a single subvolume, starting from the subvolume Green function. In the next section we will examine how these scattering matrices, which are defined on the same relatively small two-dimensional basis as the embedding potential, can help us in finding embedding potentials for multiple subvolume systems.

3.4.2 From scattering back to embedding again

In the previous section we have defined basis functions for the scattering matrices, only in the infinitesimally thick region in between subvolumes. The basis functions were defined with their z-origins to the left and right of the subvolume. Before we can stack subvolumes, we have to shift the origin of the basis functions to an origin defined in the *inter-subvolume-region.*

3.4.2.1 Shifting the origin of the scattering matrices

The situation under examination is shown in figure 3.5.

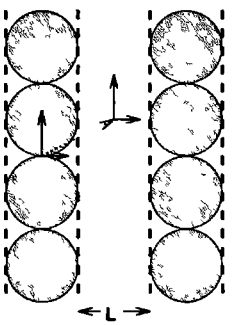


Figure 3.5: *Two subvolumes with their inter-subvolume shown exaggerated The two different origins, subvolume-origin and inter-subvolume-origin are sketched using axis.*

With several subvolumes, we have the vectors between the origins of the subvolumes, the $\mathbf{d}_n$-vectors, as shown in figure 3.6.

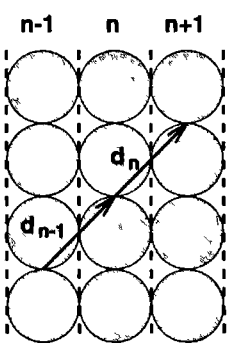


Figure 3 6: *Section of a subvolume stack. The vectors* $\mathbf{d}_{n-1}$ *and* $\mathbf{d}_n$ *indicate the shift between the subvolume origins.*

From figures 3.5 and 3.6 it is clear that the coordinate system in the left inter-subvolume of subvolume n is shifted from the origin of this subvolume by $-\frac{\mathbf{d}_{n-1}}{2}$, similarly the origin of the

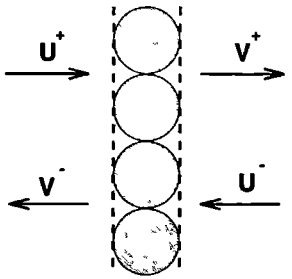


Figure 3.7: *Single subvolume with incoming waves U and outgoing waves V. Pluses and minuses indicate direction of travel.*

coordinate system in the right inter-subvolume of subvolume n is shifted from the origin of subvolume n by $\frac{\mathbf{d}_n}{2}$.
To construct the scattering matrices with respect to the correct coordinate systems, we look at the situation as shown in figure 3.7, focusing on just one subvolume, subvolume n:
We can write these waves as:

$$U^+(\mathbf{r}) = \sum_{\mathbf{G}} U^+_{\mathbf{G}} e^{i(\mathbf{k}_\parallel+\mathbf{G})\,\mathbf{R}} e^{\gamma^+_{\mathbf{G}} z} \tag{3.45a}$$

$$U^-(\mathbf{r}) = \sum_{\mathbf{G}} U^-_{\mathbf{G}} e^{i(\mathbf{k}_\parallel+\mathbf{G})\cdot\mathbf{R}} e^{\gamma^-_{\mathbf{G}} z} \tag{3.45b}$$

$$V^+(\mathbf{r}) = \sum_{\mathbf{G}} V^+_{\mathbf{G}} e^{i(\mathbf{k}_\parallel+\mathbf{G})\,\mathbf{R}} e^{\gamma^+_{\mathbf{G}} z} \tag{3.45c}$$

$$V^-(\mathbf{r}) = \sum_{\mathbf{G}} V^-_{\mathbf{G}} e^{i(\mathbf{k}_\parallel+\mathbf{G})\cdot\mathbf{R}} e^{\gamma^-_{\mathbf{G}} z} \tag{3.45d}$$

$$\tag{3.45e}$$

where V^+, V^-, U^+ and U^- are all defined with respect to the origin of subvolume n. We can write down the following relations between the amplitude coefficients of these four waves:

$$V^+_{\mathbf{G}} = \sum_{\mathbf{G}'} T^L_{\mathbf{G}\mathbf{G}'} U^+_{\mathbf{G}'} + \sum_{\mathbf{G}'} R^R_{\mathbf{G}\mathbf{G}'} U^-_{\mathbf{G}'} \tag{3.46a}$$

$$V^-_{\mathbf{G}} = \sum_{\mathbf{G}'} R^L_{\mathbf{G}\mathbf{G}'} U^+_{\mathbf{G}'} + \sum_{\mathbf{G}'} T^R_{\mathbf{G}\mathbf{G}'} U^-_{\mathbf{G}'} \tag{3.46b}$$

However, we want to define new waves with respect to the origins to the right and left of the subvolumes. First the waves on the left. We transform $\mathbf{r}$ like:

$$\mathbf{r} \mapsto \mathbf{r}'_L - \frac{\mathbf{d}_{n-1}}{2} \tag{3.47a}$$

or, in (planar, perpendicular) notation:

$$(\mathbf{R}, z) \mapsto \left(\mathbf{R}'_L - \frac{\mathbf{D}_{n-1}}{2}, z'_L - \frac{d_{n-1}}{2}\right) \tag{3.47b}$$

We have to remember that this is a coordinate transform, so we have:

$$U^+(\mathbf{r}) = U'^+(\mathbf{r}'_L) \tag{3.48}$$

for those $\mathbf{r}$ and $\mathbf{r}'_L$ pointing to the same point.
This gives:

$$\begin{aligned} U^+(\mathbf{r}) &= \sum_{\mathbf{G}} U^+_{\mathbf{G}} e^{i(\mathbf{k}_\parallel+\mathbf{G})\cdot\mathbf{R}} e^{\gamma^+_{\mathbf{G}} z} \\ &= \sum_{\mathbf{G}} U^+_{\mathbf{G}} e^{i(\mathbf{k}_\parallel+\mathbf{G})\,(\mathbf{R}'_L - \frac{\mathbf{D}_{n-1}}{2})} e^{\gamma^+_{\mathbf{G}}(z'_L - \frac{d_{n-1}}{2})} \\ &= \sum_{\mathbf{G}} U^+_{\mathbf{G}} e^{i(\mathbf{k}_\parallel+\mathbf{G})\cdot\mathbf{R}'_L} e^{\gamma^+_{\mathbf{G}} z'_L} e^{-i(\mathbf{k}_\parallel+\mathbf{G})\cdot\frac{\mathbf{D}_{n-1}}{2}} e^{-\gamma^+_{\mathbf{G}}\frac{d_{n-1}}{2}} \\ &= \sum_{\mathbf{G}} {U'_{\mathbf{G}}}^+ e^{i(\mathbf{k}_\parallel+\mathbf{G})\cdot\mathbf{R}'_L} e^{\gamma^+_{\mathbf{G}} z'_L} \end{aligned} \tag{3.49a}$$

and similarly:

$$\begin{aligned} V^-(\mathbf{r}) &= \sum_{\mathbf{G}} V^-_{\mathbf{G}} e^{i(\mathbf{k}_\parallel+\mathbf{G})\cdot\mathbf{R}} e^{\gamma^-_{\mathbf{G}} z} \\ &= \sum_{\mathbf{G}} V^-_{\mathbf{G}} e^{i(\mathbf{k}_\parallel+\mathbf{G})\cdot\mathbf{R}'_L} e^{\gamma^-_{\mathbf{G}} z'_L} e^{-i(\mathbf{k}_\parallel+\mathbf{G})\,\frac{\mathbf{D}_{n-1}}{2}} e^{-\gamma^-_{\mathbf{G}}\frac{d_{n-1}}{2}} \\ &= \sum_{\mathbf{G}} {V'_{\mathbf{G}}}^- e^{i(\mathbf{k}_\parallel+\mathbf{G})\,\mathbf{R}'_L} e^{\gamma^+_{\mathbf{G}} z'_L} \end{aligned} \tag{3.49b}$$

The same derivation holds for the waves on the right. We use the transform:

$$\mathbf{r} \mapsto \mathbf{r}'_R + \frac{\mathbf{d}_n}{2} \tag{3.50a}$$

or, in (planar, perpendicular) notation:

$$(\mathbf{R}, z) \mapsto \left(\mathbf{R}'_R + \frac{\mathbf{D}_n}{2}, z'_R + \frac{d_n}{2}\right) \tag{3.50b}$$

We then obtain:

$$\begin{aligned} V^+(\mathbf{r}) &= \sum_{\mathbf{G}} V^+_{\mathbf{G}} e^{i(\mathbf{k}_\parallel+\mathbf{G})\cdot\mathbf{R}} e^{\gamma^+_{\mathbf{G}} z} \\ &= \sum_{\mathbf{G}} V^+_{\mathbf{G}} e^{i(\mathbf{k}_\parallel+\mathbf{G})\,(\mathbf{R}'_R + \frac{\mathbf{D}_n}{2})} e^{\gamma^+_{\mathbf{G}}(z'_R + \frac{d_n}{2})} \\ &= \sum_{\mathbf{G}} V^+_{\mathbf{G}} e^{i(\mathbf{k}_\parallel+\mathbf{G})\,\mathbf{R}'_R} e^{\gamma^+_{\mathbf{G}} z'_R} e^{i(\mathbf{k}_\parallel+\mathbf{G})\,\frac{\mathbf{D}_n}{2}} e^{\gamma^+_{\mathbf{G}}\frac{d_n}{2}} \\ &= \sum_{\mathbf{G}} {V'_{\mathbf{G}}}^+ e^{i(\mathbf{k}_\parallel+\mathbf{G})\,\mathbf{R}'_R} e^{\gamma^+_{\mathbf{G}} z'_R} \end{aligned} \tag{3.51a}$$

and similarly:

$$\begin{aligned} U^-(\mathbf{r}) &= \sum_{\mathbf{G}} U^-_{\mathbf{G}} e^{i(\mathbf{k}_\parallel+\mathbf{G})\,\mathbf{R}} e^{\gamma^-_{\mathbf{G}} z} \\ &= \sum_{\mathbf{G}} U^-_{\mathbf{G}} e^{i(\mathbf{k}_\parallel+\mathbf{G})\cdot\mathbf{R}'_R} e^{\gamma^-_{\mathbf{G}} z'_R} e^{i(\mathbf{k}_\parallel+\mathbf{G})\cdot\frac{\mathbf{D}_n}{2}} e^{\gamma^-_{\mathbf{G}}\frac{d_n}{2}} \\ &= \sum_{\mathbf{G}} {U'_{\mathbf{G}}}^- e^{i(\mathbf{k}_\parallel+\mathbf{G})\,\mathbf{R}'_R} e^{\gamma^+_{\mathbf{G}} z'_R} \end{aligned} \tag{3.51b}$$

We now have the transformations for the coefficients of the waves, when we invert the relations from (3.49a), (3.49b), (3.51a) and (3.51b):

$$U_{\mathbf{G}}^{+} = e^{\imath(\mathbf{k}_{\parallel}+\mathbf{G})\frac{\mathbf{D}_{n-1}}{2}} e^{\gamma_{\mathbf{G}}^{+}\frac{d_{n-1}}{2}} U_{\mathbf{g}}'^{+} \tag{3.52a}$$

$$V_{\mathbf{G}}^{-} = e^{\imath(\mathbf{k}_{\parallel}+\mathbf{G})\frac{\mathbf{D}_{n-1}}{2}} e^{\gamma_{\mathbf{G}}^{-}\frac{d_{n-1}}{2}} V_{\mathbf{G}}'^{-} \tag{3.52b}$$

$$V_{\mathbf{G}}^{+} = e^{-\imath(\mathbf{k}_{\parallel}+\mathbf{G})\frac{\mathbf{D}_{n}}{2}} e^{-\gamma_{\mathbf{G}}^{+}\frac{d_{n}}{2}} V_{\mathbf{g}}'^{+} \tag{3.52c}$$

$$U_{\mathbf{G}}^{-} = e^{-\imath(\mathbf{k}_{\parallel}+\mathbf{G})\cdot\frac{\mathbf{D}_{n}}{2}} e^{-\gamma_{\mathbf{G}}^{-}\frac{d_{n}}{2}} U_{\mathbf{G}}'^{-} \tag{3.52d}$$

Upon substituting (3.52a) to (3.52d) into (3.46a) and (3.46b), and writing it into the form:

$$V_{\mathbf{G}}'^{+} = \sum_{\mathbf{G}'} T'^{L}_{\mathbf{G}\mathbf{G}'} U'^{+}_{\mathbf{G}'} + \sum_{\mathbf{G}'} R'^{R}_{\mathbf{G}\mathbf{G}'} U'^{-}_{\mathbf{G}'} \tag{3.53a}$$

$$V_{\mathbf{G}}'^{-} = \sum_{\mathbf{G}'} R'^{L}_{\mathbf{G}\mathbf{G}'} U'^{+}_{\mathbf{G}'} + \sum_{\mathbf{G}'} T'^{R}_{\mathbf{G}\mathbf{G}'} U'^{-}_{\mathbf{G}'} \tag{3.53b}$$

we get the definitions:

$$T'^{L}_{\mathbf{G}\mathbf{G}'} = e^{\imath(\mathbf{k}_{\parallel}+\mathbf{G})\frac{\mathbf{D}_{n}}{2}} e^{\gamma_{\mathbf{G}}^{+}\frac{d_{n}}{2}} T^{L}_{\mathbf{G}\mathbf{G}'} e^{\imath(\mathbf{k}_{\parallel}+\mathbf{G}')\frac{\mathbf{D}_{n-1}}{2}} e^{\gamma_{\mathbf{G}'}^{+}\frac{d_{n-1}}{2}} \tag{3.54a}$$

$$R'^{R}_{\mathbf{G}\mathbf{G}'} = e^{\imath(\mathbf{k}_{\parallel}+\mathbf{G})\frac{\mathbf{D}_{n}}{2}} e^{\gamma_{\mathbf{G}}^{+}\frac{d_{n}}{2}} R^{R}_{\mathbf{G}\mathbf{G}'} e^{-\imath(\mathbf{k}_{\parallel}+\mathbf{G}')\cdot\frac{\mathbf{D}_{n}}{2}} e^{-\gamma_{\mathbf{G}'}^{-}\frac{d_{n}}{2}} \tag{3.54b}$$

$$R'^{L}_{\mathbf{G}\mathbf{G}'} = e^{-\imath(\mathbf{k}_{\parallel}+\mathbf{G})\cdot\frac{\mathbf{D}_{n-1}}{2}} e^{-\gamma_{\mathbf{G}}^{-}\frac{d_{n-1}}{2}} R^{L}_{\mathbf{G}\mathbf{G}'} e^{\imath(\mathbf{k}_{\parallel}+\mathbf{G}')\cdot\frac{\mathbf{D}_{n-1}}{2}} e^{\gamma_{\mathbf{G}'}^{+}\frac{d_{n-1}}{2}} \tag{3.54c}$$

$$T'^{R}_{\mathbf{G}\mathbf{G}'} = e^{-\imath(\mathbf{k}_{\parallel}+\mathbf{G})\cdot\frac{\mathbf{D}_{n-1}}{2}} e^{-\gamma_{\mathbf{G}}^{-}\frac{d_{n-1}}{2}} T^{R}_{\mathbf{G}\mathbf{G}'} e^{-\imath(\mathbf{k}_{\parallel}+\mathbf{G}')\cdot\frac{\mathbf{D}_{n}}{2}} e^{-\gamma_{\mathbf{G}'}^{-}\frac{d_{n}}{2}} \tag{3.54d}$$

So now we know how to transform between reflection and transmission matrices in terms of the subvolume-origin basis, and reflection and transmission matrices in terms of the inter-subvolume-origin basis.

I now want to introduce a new matrix, which I will call the transfer matrix. This matrix relates waves on one side of a system to waves on the other side.

3.4.2.2 Introducing the transfer matrix $\mathbb{T}$

Let us concentrate on a single subvolume first, the one shown in figure 3.7. We want to know how the waves on the left relate to the ones on the right. To do this we will construct an operator called the right-to-left transfer matrix. In this entire derivation we give the wave-coefficients and scattering matrices an index for the subvolume they belong to. We want to reach the following expression:

$$\begin{pmatrix} U_n'^{+} \\ V_n'^{-} \end{pmatrix} = \mathbb{T}_n^{\leftarrow} \times \begin{pmatrix} V_n'^{+} \\ U_n'^{-} \end{pmatrix} \tag{3.55}$$

where $\mathbb{T}_n^{\leftarrow}$ is the right-to-left transfer matrix for subvolume n. We construct this transfer matrix from equations (3.53a) and (3.53b). First we rewrite (3.53a):

$$U_n'^{+} = -(T_n'^{L})^{-1} \times \left(R_n'^{R} U_n'^{-} - V_n'^{+} \right) \tag{3.56}$$

When we substitute (3.56) into (3.53b) we then obtain:

$$V_n'^{-} = -R_n'^{L} \times (T_n'^{L})^{-1} \times \left(R_n'^{R} U_n'^{-} - V_n'^{+} \right) + T_n'^{R} U_n'^{-} \tag{3.57}$$

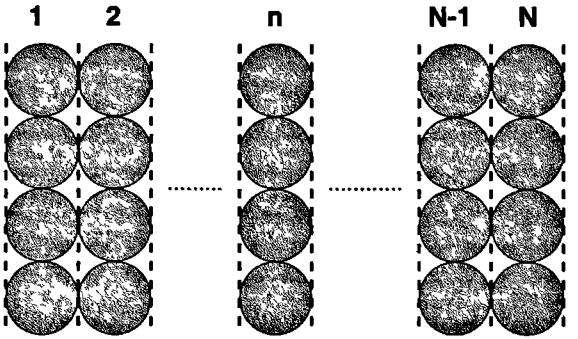


Figure 3.8: *Stack of N subvolumes. We want to find reflection and transmission properties for the entire stack.*

or, collecting terms:

$$U_n'^{+} = (T_n'^{L})^{-1} V_n'^{+} - (T_n'^{L})^{-1} R_n'^{R} U_n'^{-} \tag{3.58a}$$

$$V_n'^{-} = R_n'^{L} (T_n'^{L})^{-1} V_n'^{+} + \left(T_n'^{R} - R_n'^{L} (T_n'^{L})^{-1} R_n'^{R} \right) U_n'^{-} \tag{3.58b}$$

Writing this in a matrix form similar to (3.55), we get:

$$\begin{pmatrix} U_n'^{+} \\ V_n'^{-} \end{pmatrix} = \begin{pmatrix} (T_n'^{L})^{-1} & -(T_n'^{L})^{-1} R_n'^{R} \\ R_n'^{L} (T_n'^{L})^{-1} & \left(T_n'^{R} - R_n'^{L} (T_n'^{L})^{-1} R_n'^{R} \right) \end{pmatrix} \times \begin{pmatrix} V_n'^{+} \\ U_n'^{-} \end{pmatrix} \tag{3.59}$$

This gives us the matrix expression for the right-to-left transfer matrix:

$$\mathbb{T}_n^{\leftarrow} = \begin{pmatrix} (T_n'^{L})^{-1} & -(T_n'^{L})^{-1} R_n'^{R} \\ R_n'^{L} (T_n'^{L})^{-1} & \left(T_n'^{R} - R_n'^{L} (T_n'^{L})^{-1} R_n'^{R} \right) \end{pmatrix} \tag{3.60}$$

Using the definition:

$$\begin{pmatrix} V_n'^{+} \\ U_n'^{-} \end{pmatrix} = \mathbb{T}_n^{\rightarrow} \times \begin{pmatrix} U_n'^{+} \\ V_n'^{-} \end{pmatrix} \tag{3.61}$$

we can similarly construct an expression for $\mathbb{T}_n^{\rightarrow}$, the left-to-right transfer matrix:

$$\mathbb{T}_n^{\rightarrow} = \begin{pmatrix} \left(T_n'^{L} - R_n'^{R} (T_n'^{R})^{-1} R_n'^{L} \right) & R_n'^{R} (T_n'^{R})^{-1} \\ -(T_n'^{R})^{-1} R_n'^{L} & (T_n'^{R})^{-1} \end{pmatrix} \tag{3.62}$$

3.4.2.3 Scattering of multiple subvolumes

It turns out that the transfer matrix is an excellent operator for finding the scattering properties of a stack of subvolumes. Let us suppose we have a stack of N subvolumes, as shown in figure 3.8, and we want to construct scattering properties for this stack. We can start by having waves on the right of the stack, in both the + and − direction. We call their

amplitudes ${V'_N}^+$ and ${U'_N}^-$. We construct the waves in between subvolumes $N-1$ and N by multiplying the waves ${V'_N}^+$ and ${U'_N}^-$ with the right to left transfer matrix for subvolume N.

$$\begin{pmatrix} {V'_{N-1}}^+ \\ {U'_{N-1}}^- \end{pmatrix} = \begin{pmatrix} {U'_N}^+ \\ {V'_N}^- \end{pmatrix} = \mathbb{T}^{\leftarrow}_N \times \begin{pmatrix} {V'_N}^+ \\ {U'_N}^- \end{pmatrix} \tag{3.63}$$

The first equality in (3.63) follows because the origins of the coordinate systems of both coincide. Now from (3.63) it becomes obvious what the next step is: we take the vector in the first term through subvolume $N-1$ by multiplying with the transfer matrix for that subvolume, and we can repeat that process all the way through the stack. So finally we get:

$$\begin{pmatrix} U'^+_1 \\ V'^-_1 \end{pmatrix} = \mathbb{T}^{\leftarrow}_1 \times \mathbb{T}^{\leftarrow}_2 \times \cdots \times \mathbb{T}^{\leftarrow}_{N-1} \times \mathbb{T}^{\leftarrow}_N \times \begin{pmatrix} {V'_N}^+ \\ {U'_N}^- \end{pmatrix} \tag{3.64}$$

Hence the transfer matrix for the entire stack is given by:

$$\mathbb{T}^{\leftarrow}_{1\cdots N} = \mathbb{T}^{\leftarrow}_1 \times \mathbb{T}^{\leftarrow}_2 \times \cdots \times \mathbb{T}^{\leftarrow}_{N-1} \times \mathbb{T}^{\leftarrow}_N \tag{3.65}$$

In a similar way we construct $\mathbb{T}^{\rightarrow}_{1\cdots N}$ defined by:

$$\begin{pmatrix} {U'_N}^- \\ {V'_N}^+ \end{pmatrix} = \mathbb{T}^{\rightarrow}_{1\cdots N} \times \begin{pmatrix} V'^-_1 \\ U'^+_1 \end{pmatrix} \tag{3.66}$$

where

$$\mathbb{T}^{\rightarrow}_{1\cdots N} = \mathbb{T}^{\rightarrow}_N \times \mathbb{T}^{\rightarrow}_{N-1} \times \cdots \times \mathbb{T}^{\rightarrow}_2 \times \mathbb{T}^{\rightarrow}_1 \tag{3.67}$$

Splitting these transfer matrices up into sub-matrices as defined in (3.60) and (3.62), we can finally derive the R and T matrices for the whole stack.

3.4.2.4 Scattering of a semi-infinite system

We have seen in section 3.3 that we often need to construct the embedding potential for a semi-infinite substrate. This can be done from the reflection matrix of the semi-infinite substrate, which we shall now show how to construct.

First, let us look at the situation described in figure 3.9.

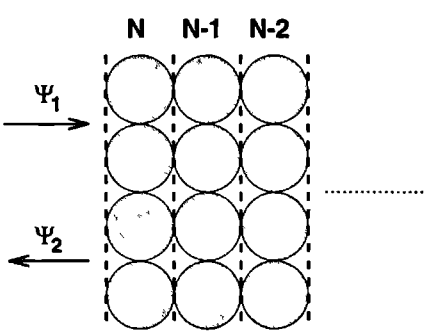


Figure 3.9: *Stack of N subvolumes.*

We will assume that N is big enough so that the reflection does not change significantly upon adding one more layer. This leads to the following relations between Ψ_1 and Ψ_2.

$$\Psi_2 = R_N \Psi_1 \tag{3.68}$$

Now let us add one more layer to the stack, as shown in figure 3.10.

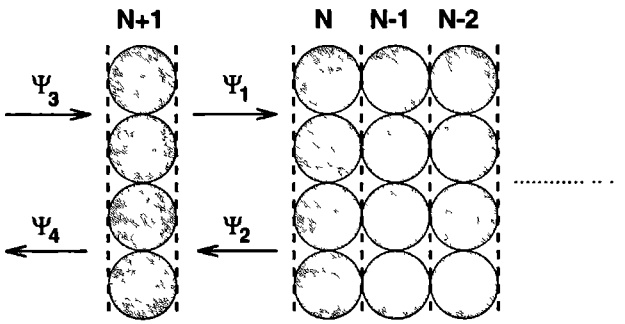


Figure 3.10: *Stack of $N+1$ subvolumes. The inter-subvolume between subvolume N and $N+1$ has been exaggerated for clarity.*

We can now relate Ψ_3 and Ψ_4 to Ψ_1 and Ψ_2 through the transfer matrix:

$$\begin{pmatrix} \Psi_3 \\ \Psi_4 \end{pmatrix} = \begin{pmatrix} \mathbb{T}_{11}^{\leftarrow} & \mathbb{T}_{12}^{\leftarrow} \\ \mathbb{T}_{21}^{\leftarrow} & \mathbb{T}_{22}^{\leftarrow} \end{pmatrix} \times \begin{pmatrix} \Psi_1 \\ \Psi_2 \end{pmatrix} \tag{3.69}$$

We have to remember that Ψ_3 and Ψ_4 are related through:

$$\Psi_4 = R_{N+1} \Psi_3 \tag{3.70}$$

Writing (3.69) in the following form:

$$\Psi_3 = \mathbb{T}_{11}^{\leftarrow} \Psi_1 + \mathbb{T}_{12}^{\leftarrow} \Psi_2 \tag{3.71a}$$

$$\Psi_4 = \mathbb{T}_{21}^{\leftarrow} \Psi_1 + \mathbb{T}_{22}^{\leftarrow} \Psi_2 \tag{3.71b}$$

and using (3.68) this becomes:

$$\Psi_3 = (\mathbb{T}_{11}^{\leftarrow} + \mathbb{T}_{12}^{\leftarrow} R_N)\, \Psi_1 \tag{3.72a}$$

$$\Psi_4 = (\mathbb{T}_{21}^{\leftarrow} + \mathbb{T}_{22}^{\leftarrow} R_N)\, \Psi_1 \tag{3.72b}$$

Inverting (3.72a) and substituting into (3.72b) gives:

$$\Psi_4 = (\mathbb{T}_{21}^{\leftarrow} + \mathbb{T}_{22}^{\leftarrow} R_N) \times (\mathbb{T}_{11}^{\leftarrow} + \mathbb{T}_{12}^{\leftarrow} R_N)^{-1}\, \Psi_3 \tag{3.73}$$

So now we have the following expression (using (3.70)):

$$R_{N+1} = (\mathbb{T}_{21}^{\leftarrow} + \mathbb{T}_{22}^{\leftarrow} R_N) \times (\mathbb{T}_{11}^{\leftarrow} + \mathbb{T}_{12}^{\leftarrow} R_N)^{-1} \tag{3.74}$$

We have again found a recursion relation for R_n. Iterating (3.74) will eventually give us the reflection matrix of the semi-infinite substrate.
Straightforward iteration of (3.74) corresponds to building the semi-infinite substrate by stacking one subvolume at a time. We can speed this up by using a Newton-Raphson like method, as discussed before in section 3.3.1.

3.4.2.5 Newton-Raphson method to stack subvolumes

Normal direct stacking of layers to find the reflection matrix for the semi-infinite system converges quite slowly. Upon examining faster methods of convergence, we can write down a function for which the physical root is the reflection matrix of the semi-infinite system.

If we first look at the semi-infinite system extending to the left, we know that the reflection matrix for the system with $N+1$ layers can be expressed in terms of the reflection matrix for the system with N layers; see (3.74). I define the multi-dimensional function, F, as:

$$F_N = R_N - (\mathbb{T}^{\leftarrow}_{21} + \mathbb{T}^{\leftarrow}_{22} R_N) \times (\mathbb{T}^{\leftarrow}_{11} + \mathbb{T}^{\leftarrow}_{12} R_N)^{-1} \tag{3.75}$$

We assume that the stack of $N+1$ subvolumes is the converged semi-infinite system, which means that:

$$F_{N+1} = 0 \tag{3.76}$$

We want an estimate for R_{N+1} from R_N, which will lead to F_{N+1} being zero. Let us write:

$$R_{N+1} = R_N - D_N, \tag{3.77}$$

then we can write down F_{N+1} as:

$$\begin{aligned}
F_{N+1} &= R_{N+1} - (\mathbb{T}^{\leftarrow}_{21} + \mathbb{T}^{\leftarrow}_{22} R_{N+1}) \times (\mathbb{T}^{\leftarrow}_{11} + \mathbb{T}^{\leftarrow}_{12} R_{N+1})^{-1} \\
&= (R_N - D_N) - (\mathbb{T}^{\leftarrow}_{21} + \mathbb{T}^{\leftarrow}_{22} (R_N - D_N)) \times (\mathbb{T}^{\leftarrow}_{11} + \mathbb{T}^{\leftarrow}_{12} (R_N - D_N))^{-1} \\
&= (R_N - D_N) - (\mathbb{T}^{\leftarrow}_{21} + \mathbb{T}^{\leftarrow}_{22} R_N - \mathbb{T}^{\leftarrow}_{22} D_N) \times (\mathbb{T}^{\leftarrow}_{11} + \mathbb{T}^{\leftarrow}_{12} R_N - \mathbb{T}^{\leftarrow}_{12} D_N)^{-1} \\
&= (R_N - D_N) - (\mathbb{T}^{\leftarrow}_{21} + \mathbb{T}^{\leftarrow}_{22} R_N - \mathbb{T}^{\leftarrow}_{22} D_N) \times \\
&\qquad \left[(\mathbb{T}^{\leftarrow}_{11} + \mathbb{T}^{\leftarrow}_{12} R_N) \left(1 - (\mathbb{T}^{\leftarrow}_{11} + \mathbb{T}^{\leftarrow}_{12} R_N)^{-1} \mathbb{T}^{\leftarrow}_{12} D_N \right) \right]^{-1} \\
&= (R_N - D_N) - (\mathbb{T}^{\leftarrow}_{21} + \mathbb{T}^{\leftarrow}_{22} R_N - \mathbb{T}^{\leftarrow}_{22} D_N) \times \\
&\qquad \left[\left(1 - (\mathbb{T}^{\leftarrow}_{11} + \mathbb{T}^{\leftarrow}_{12} R_N)^{-1} \mathbb{T}^{\leftarrow}_{12} D_N \right)^{-1} (\mathbb{T}^{\leftarrow}_{11} + \mathbb{T}^{\leftarrow}_{12} R_N)^{-1} \right]
\end{aligned} \tag{3.78}$$

Using a Taylor expansion to first order in D_N, yields:

$$\begin{aligned}
F_{N+1} = (R_N - D_N) &- (\mathbb{T}^{\leftarrow}_{21} + \mathbb{T}^{\leftarrow}_{22} R_N - \mathbb{T}^{\leftarrow}_{22} D_N) \times \\
&\left[\left(1 + (\mathbb{T}^{\leftarrow}_{11} + \mathbb{T}^{\leftarrow}_{12} R_N)^{-1} \mathbb{T}^{\leftarrow}_{12} D_N \right) (\mathbb{T}^{\leftarrow}_{11} + \mathbb{T}^{\leftarrow}_{12} R_N)^{-1} \right]
\end{aligned} \tag{3.79}$$

We write this out to first order in D_N and we obtain:

$$\begin{aligned}
F_{N+1} = R_N - D_N - (\mathbb{T}^{\leftarrow}_{21} + \mathbb{T}^{\leftarrow}_{22} R_N) &\times (\mathbb{T}^{\leftarrow}_{11} + \mathbb{T}^{\leftarrow}_{12} R_N)^{-1} \\
&+ \mathbb{T}^{\leftarrow}_{22} D_N (\mathbb{T}^{\leftarrow}_{11} + \mathbb{T}^{\leftarrow}_{12} R_N)^{-1} \\
&- (\mathbb{T}^{\leftarrow}_{21} + \mathbb{T}^{\leftarrow}_{22} R_N) \times (\mathbb{T}^{\leftarrow}_{11} + \mathbb{T}^{\leftarrow}_{12} R_N)^{-1} \mathbb{T}^{\leftarrow}_{12} D_N (\mathbb{T}^{\leftarrow}_{11} + \mathbb{T}^{\leftarrow}_{12} R_N)^{-1}
\end{aligned} \tag{3.80}$$

Now using the fact that F_{N+1} is zero and (3.75), we get:

$$\begin{aligned}
F_N = D_N - \left[\mathbb{T}^{\leftarrow}_{22} - (\mathbb{T}^{\leftarrow}_{21} + \mathbb{T}^{\leftarrow}_{22} R_N) \times (\mathbb{T}^{\leftarrow}_{11} + \mathbb{T}^{\leftarrow}_{12} R_N)^{-1} \mathbb{T}^{\leftarrow}_{12} \right] &\times \\
D_N \times (\mathbb{T}^{\leftarrow}_{11} + \mathbb{T}^{\leftarrow}_{12} R_N)^{\ 1}&
\end{aligned} \tag{3.81}$$

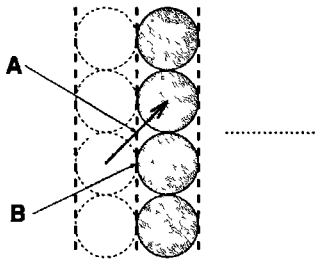


Figure 3.11: *The stack extends to 'infinity' towards the right. Points A and B indicate two different origins for the basis expanding the reflection matrix of the semi-infinite stack.*

This can be written as:

$$F_N = D_N - X_N D_N Y_N \tag{3.82}$$
$$X_N = [\mathbb{T}^{\leftarrow}_{22} - R_{N+1}\mathbb{T}^{\leftarrow}_{12}] \tag{3.83}$$
$$Y_N = (\mathbb{T}^{\leftarrow}_{11} + \mathbb{T}^{\leftarrow}_{12}R_N)^{-1} \tag{3.84}$$

where (3.82) may be solved by diagonalising X_N and Y_N[7].
Similarly the result for the semi-infinite system extending to the right, is:

$$F_N = D_N - X_N D_N Y_N \tag{3.85}$$
$$X_N = [\mathbb{T}^{\rightarrow}_{22} - R_{N+1}\mathbb{T}^{\rightarrow}_{12}] \tag{3.86}$$
$$Y_N = (\mathbb{T}^{\rightarrow}_{11} + \mathbb{T}^{\rightarrow}_{12}R_N)^{-1} \tag{3.87}$$

Upon implementation, it turns out that this Newton-Raphson stacking of subvolumes to get the reflection matrix for the semi-infinite system (and thus the embedding potential) is several orders of magnitude faster than the Green function stacking method described in section 3.3 (typically about 1000 to 2000 times faster).

3.4.2.6 From reflection matrix to embedding potential

All that remains now is to construct the embedding potential from the reflection matrix for the semi-infinite system, found using the method described in the previous section.

The reflection matrices with respect to origin B in figure 3.11, appropriate to the embedding potential are related to those with respect to origin A by:

$$R'^{L}_{\infty,\mathbf{GG}'} = e^{-\imath(\mathbf{k}_\parallel+\mathbf{G})\frac{\mathbf{D}_n}{2}} R^{L}_{\infty,\mathbf{GG}'} e^{\imath(\mathbf{k}_\parallel+\mathbf{G}')\frac{\mathbf{D}_n}{2}} \tag{3.88}$$

The derivation is similar to (3.54b). Similarly, the reflection of the right of a left semi-infinite stack, is:

$$R'^{R}_{\infty,\mathbf{GG}'} = e^{\imath(\mathbf{k}_\parallel+\mathbf{G})\frac{\mathbf{D}_{n-1}}{2}} R^{R}_{\infty,\mathbf{GG}'} e^{-\imath(\mathbf{k}_\parallel+\mathbf{G}')\frac{\mathbf{D}_{n-1}}{2}} \tag{3.89}$$

Again this derivation is similar to (3.54b).

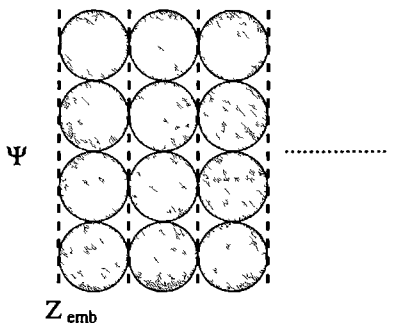


Figure 3.12: *The stack extends to infinity towards the right. Z_{emb} is the z-coordinate of the embedding plane in the coordinates of the left-most subvolume.*

Suppose we have a reflection matrix R^L_∞ at the left of a right semi-infinite stack of subvolumes, represented in the appropriate basis (notice that I have dropped the prime on the reflection matrix, but is the reflection matrix with respect to origin B!). The situation is sketched in figure 3.12. The basis functions will be represented as $\phi^\pm_{\mathbf{G}}(\mathbf{r})$, where plus or minus indicates the direction in which the waves move. We can write Ψ as:

$$\Psi(\mathbf{r}) = \sum_{\mathbf{G}} a_{\mathbf{G}} \phi^+_{\mathbf{G}}(\mathbf{r}) + \sum_{\mathbf{G}} \left(\sum_{\mathbf{G}'} R^L_{\infty,\mathbf{G}\mathbf{G}'} a_{\mathbf{G}'} \right) \phi^-_{\mathbf{G}}(\mathbf{r}) \tag{3.90a}$$

and, switching $\mathbf{G}$ and $\mathbf{G}'$ indices on the last term, this becomes:

$$\Psi(\mathbf{r}) = \sum_{\mathbf{G}} a_{\mathbf{G}} \left(\phi^+_{\mathbf{G}}(\mathbf{r}) + \sum_{\mathbf{G}'} R^L_{\infty,\mathbf{G}'\mathbf{G}} \phi^-_{\mathbf{G}'}(\mathbf{r}) \right) \tag{3.90b}$$

with derivative:

$$\frac{\partial \Psi(\mathbf{r})}{\partial n_S} = \sum_{\mathbf{G}} a_{\mathbf{G}} \left(\frac{\partial \phi^+_{\mathbf{G}}(\mathbf{r})}{\partial n_S} + \sum_{\mathbf{G}'} R^L_{\infty,\mathbf{G}'\mathbf{G}} \frac{\partial \phi^-_{\mathbf{G}'}(\mathbf{r})}{\partial n_S} \right) \tag{3.91}$$

Here the basis functions and their derivatives are given by:

$$\phi^\pm_{\mathbf{G}}(\mathbf{r}) = \frac{1}{\sqrt{A}} e^{i(\mathbf{k}_\parallel + \mathbf{G})\,\mathbf{R}} e^{\gamma^\pm_{\mathbf{G}} z} \tag{3.92a}$$

$$\frac{\partial \phi^\pm_{\mathbf{G}}(\mathbf{r})}{\partial n_S} = \frac{1}{\sqrt{A}} \gamma^\pm_{\mathbf{G}} e^{i(\mathbf{k}_\parallel + \mathbf{G})\,\mathbf{R}} e^{\gamma^\pm_{\mathbf{G}} z} = \gamma^\pm_{\mathbf{G}} \phi^\pm_{\mathbf{G}}(\mathbf{r}) \tag{3.92b}$$

The wavefunction and its derivative are related through the embedding potential in the following way:

$$\frac{\partial \Psi(\mathbf{R}_S)}{\partial n_S} = -2 \int_S \mathrm{d}^2\mathbf{R}'_S \Sigma(\mathbf{R}_S, \mathbf{R}'_S) \Psi(\mathbf{R}'_S) \tag{3.93}$$

We substitute (3.90b) and (3.91) into (3.93) and we obtain:

$$\sum_{\mathbf{G}} a_{\mathbf{G}} \left(\gamma_{\mathbf{G}}^{+} \phi_{\mathbf{G}}^{+}(\mathbf{R}_S) + \sum_{\mathbf{G}'} R^{L}_{\infty,\mathbf{G}'\mathbf{G}} \gamma_{\mathbf{G}'}^{-} \phi_{\mathbf{G}'}^{-}(\mathbf{R}_S) \right) = - 2 \int_S \mathrm{d}^2\mathbf{R}'_S \Sigma(\mathbf{R}_S, \mathbf{R}'_S) \sum_{\mathbf{G}} a_{\mathbf{G}} \left(\phi_{\mathbf{G}}^{+}(\mathbf{R}'_S) + \sum_{\mathbf{G}'} R^{L}_{\infty,\mathbf{G}'\mathbf{G}} \phi_{\mathbf{G}'}^{-}(\mathbf{R}'_S) \right) \quad (3.94)$$

We define the planar basis functions:

$$e_{\mathbf{G}}(\mathbf{R}_S) = \frac{1}{\sqrt{A}} e^{i(\mathbf{k}_{\parallel}+\mathbf{G})\cdot\mathbf{R}_S} \quad (3.95)$$

and write the embedding potential in terms of these:

$$\Sigma(\mathbf{R}_S, \mathbf{R}'_S) = \sum_{\mathbf{G}\mathbf{G}'} e_{\mathbf{G}}(\mathbf{R}_S) \Sigma_{\mathbf{G}\mathbf{G}'} e^{*}_{\mathbf{G}'}(\mathbf{R}'_S) \quad (3.96)$$

We use the following relations:

$$\int_S \mathrm{d}^2\mathbf{R}_S e^{*}_{\mathbf{G}}(\mathbf{R}_S) e_{\mathbf{G}'}(\mathbf{R}_S) = \delta_{\mathbf{G}\mathbf{G}'} \quad (3.97a)$$

$$\int_S \mathrm{d}^2\mathbf{R}_S e^{*}_{\mathbf{G}}(\mathbf{R}_S) \phi^{\pm}_{\mathbf{G}'}(\mathbf{R}_S) = \delta_{\mathbf{G}\mathbf{G}'} e^{\gamma^{\pm}_{\mathbf{G}'} z_S} \quad (3.97b)$$

and can now write down the following equation for a single $\mathbf{G}$ term of (3.94):

$$\left(\gamma_{\mathbf{G}}^{+} \phi_{\mathbf{G}}^{+}(\mathbf{R}_S) + \sum_{\mathbf{G}'} R^{L}_{\infty,\mathbf{G}'\mathbf{G}} \gamma_{\mathbf{G}'}^{-} \phi_{\mathbf{G}'}^{-}(\mathbf{R}_S) \right) = - 2 \sum_{\mathbf{G}_2\mathbf{G}_3} e_{\mathbf{G}_2}(\mathbf{R}_S) \Sigma_{\mathbf{G}_2\mathbf{G}_3} \int_S \mathrm{d}^2\mathbf{R}'_S e^{*}_{\mathbf{G}_3}(\mathbf{R}'_S) \phi_{\mathbf{G}}^{+}(\mathbf{R}'_S) - 2 \sum_{\mathbf{G}_2\mathbf{G}_3} e_{\mathbf{G}_2}(\mathbf{R}_S) \Sigma_{\mathbf{G}_2\mathbf{G}_3} \sum_{\mathbf{G}'} R^{L}_{\infty,\mathbf{G}'\mathbf{G}} \int_S \mathrm{d}^2\mathbf{R}'_S e^{*}_{\mathbf{G}_3}(\mathbf{R}'_S) \phi_{\mathbf{G}'}^{-}(\mathbf{R}'_S) \quad (3.98)$$

Using (3.97a) and (3.97b) this becomes:

$$\left(\gamma_{\mathbf{G}}^{+} \phi_{\mathbf{G}}^{+}(\mathbf{R}_S) + \sum_{\mathbf{G}'} R^{L}_{\infty,\mathbf{G}'\mathbf{G}} \gamma_{\mathbf{G}'}^{-} \phi_{\mathbf{G}'}^{-}(\mathbf{R}_S) \right) = - 2 \sum_{\mathbf{G}_2\mathbf{G}_3} e_{\mathbf{G}_2}(\mathbf{R}_S) \Sigma_{\mathbf{G}_2\mathbf{G}_3} \delta_{\mathbf{G}_3\mathbf{G}} e^{\gamma_{\mathbf{G}}^{+} z'_S} - 2 \sum_{\mathbf{G}_2\mathbf{G}_3} e_{\mathbf{G}_2}(\mathbf{R}_S) \Sigma_{\mathbf{G}_2\mathbf{G}_3} \sum_{\mathbf{G}'} R^{L}_{\infty,\mathbf{G}'\mathbf{G}} \delta_{\mathbf{G}_3\mathbf{G}'} e^{\gamma_{\mathbf{G}'}^{-} z'_S} \quad (3.99)$$

Multiplying from the left with $e^{*}_{\mathbf{G}_1}(\mathbf{R}_S)$ and integrating $\mathbf{R}_S$ over S, we get:

$$\int_S \mathrm{d}^2\mathbf{R}_S e^{*}_{\mathbf{G}_1}(\mathbf{R}_S) \left(\gamma_{\mathbf{G}}^{+} \phi_{\mathbf{G}}^{+}(\mathbf{R}_S) + \sum_{\mathbf{G}'} R^{L}_{\infty,\mathbf{G}'\mathbf{G}} \gamma_{\mathbf{G}'}^{-} \phi_{\mathbf{G}'}^{-}(\mathbf{R}_S) \right) = - 2 \sum_{\mathbf{G}_2\mathbf{G}_3} \int_S \mathrm{d}^2\mathbf{R}_S e^{*}_{\mathbf{G}_1}(\mathbf{R}_S) e_{\mathbf{G}_2}(\mathbf{R}_S) \Sigma_{\mathbf{G}_2\mathbf{G}_3} \sum_{\mathbf{G}'} R^{L}_{\infty,\mathbf{G}'\mathbf{G}} \delta_{\mathbf{G}_3\mathbf{G}'} e^{\gamma_{\mathbf{G}'}^{-} z'_S} \quad (3.100)$$

which becomes:

$$\gamma^+_{\mathbf{G}} e^{\gamma^+_{\mathbf{G}} z_S} \delta_{\mathbf{G}_1\mathbf{G}} + \sum_{\mathbf{G}'} R^L_{\infty,\mathbf{G}'\mathbf{G}} \gamma^-_{\mathbf{G}'} e^{\gamma^-_{\mathbf{G}'} z_S} \delta_{\mathbf{G}_1\mathbf{G}'} =$$

$$-2 \sum_{\mathbf{G}_2\mathbf{G}_3} \delta_{\mathbf{G}_1\mathbf{G}_2} \Sigma_{\mathbf{G}_2\mathbf{G}_3} \delta_{\mathbf{G}_3\mathbf{G}} e^{\gamma^+_{\mathbf{G}} z'_S} - 2 \sum_{\mathbf{G}_2\mathbf{G}_3} \delta_{\mathbf{G}_1\mathbf{G}_2} \Sigma_{\mathbf{G}_2\mathbf{G}_3} \sum_{\mathbf{G}'} R^L_{\infty,\mathbf{G}'\mathbf{G}} \delta_{\mathbf{G}_3\mathbf{G}'} e^{\gamma^-_{\mathbf{G}'} z'_S} \quad (3.101)$$

and this simplifies to:

$$\gamma^+_{\mathbf{G}_1} e^{\gamma^+_{\mathbf{G}_1} z_S} + R^L_{\infty,\mathbf{G}_1\mathbf{G}} \gamma^-_{\mathbf{G}_1} e^{\gamma^-_{\mathbf{G}_1} z_S} = -2\Sigma_{\mathbf{G}_1\mathbf{G}} e^{\gamma^+_{\mathbf{G}} z'_S} - 2 \sum_{\mathbf{G}'} \Sigma_{\mathbf{G}_1\mathbf{G}'} R^L_{\infty,\mathbf{G}'\mathbf{G}} e^{\gamma^-_{\mathbf{G}'} z'_S} \quad (3.102)$$

Now we use the fact that the origin is on the embedding plane position, and therefore z_S and z'_S are zero. We then get:

$$\gamma^+_{\mathbf{G}_1} + R^L_{\infty,\mathbf{G}_1\mathbf{G}} \gamma^-_{\mathbf{G}_1} = -2\Sigma_{\mathbf{G}_1\mathbf{G}} - 2 \sum_{\mathbf{G}'} \Sigma_{\mathbf{G}_1\mathbf{G}'} R^L_{\infty,\mathbf{G}'\mathbf{G}} \quad (3.103)$$

Now let us define the matrix M in the following way:

$$M_{\mathbf{G}\mathbf{G}'} = \gamma^+_{\mathbf{G}} \delta_{\mathbf{G}\mathbf{G}'} + \gamma^-_{\mathbf{G}} R^L_{\infty,\mathbf{G}\mathbf{G}'} \quad (3.104)$$

then (3.103) becomes in matrix form:

$$\Sigma \left(1 + R^L_\infty\right) = -\tfrac{1}{2} M \quad (3.105)$$

or:

$$\Sigma = -\tfrac{1}{2} M \left(1 + R^L_\infty\right)^{-1} \quad (3.106)$$

and in explicit notation:

$$\Sigma_{\mathbf{G}\mathbf{G}'} = -\tfrac{1}{2} \sum_{\mathbf{G}_1} \left(\gamma^+_{\mathbf{G}} \delta_{\mathbf{G}\mathbf{G}_1} + \gamma^-_{\mathbf{G}} R^L_{\infty,\mathbf{G}\mathbf{G}_1}\right) \left(1 + R^L_\infty\right)^{-1}_{\mathbf{G}_1\mathbf{G}'}. \quad (3.107)$$

Using $\gamma^- = -\gamma^+$ this becomes:

$$\Sigma_{\mathbf{G}\mathbf{G}'} = -\tfrac{1}{2} \gamma^+_{\mathbf{G}} \sum_{\mathbf{G}_1} \left(\delta_{\mathbf{G}\mathbf{G}_1} - R^L_{\infty,\mathbf{G}\mathbf{G}_1}\right) \left(1 + R^L_\infty\right)^{-1}_{\mathbf{G}_1\mathbf{G}'} \quad (3.108)$$

Here in (3.108) we have an expression for the embedding potential of the semi-infinite stack of subvolumes in terms of its reflection matrix, which we can find from (3.75). A similar expression has been derived before[3] for a reflection matrix corresponding to a muffin-tin substrate.

A similar derivation can be performed for the left-embedding potential of a left semi-infinite stack of subvolumes and eventually yields:

$$\Sigma_{\mathbf{G}\mathbf{G}'} = -\tfrac{1}{2} \gamma^+_{\mathbf{G}} \sum_{\mathbf{G}_1} \left(\delta_{\mathbf{G}\mathbf{G}_1} - R^R_{\infty,\mathbf{G}\mathbf{G}_1}\right) \left(1 + R^R_\infty\right)^{-1}_{\mathbf{G}_1\mathbf{G}'} \quad (3.109)$$

We see that this has the same form as (3.108), with R^R substituted for R^L.

We now have a working efficient algorithm for finding the embedding potential for a semi-infinite stack of similar subvolumes. The only remaining step is to write the propagation of embedding potentials through subvolumes in terms of scattering properties.

3.4.2.7 Propagating embedding potentials through scattering

There are actually two ways to do this. The first is relatively simple, since we already have all the ingredients, in sections 3.4.1, 3.4.2.2, 3.4.2.3 and 3.4.2.6. A rough outline of this process goes like this, keeping in mind figure 3.1:

- start with embedding potentials **and** reflection matrices on S_R and S_L.
- calculate reflection matrix of left substrate plus subvolume 1.
- use this to calculate left embedding potential on S_1.
- calculate reflection matrix of left substrate plus subvolumes 1 and 2.
- use this to calculate left embedding potential on S_2.
- repeat until all left embedding potentials are found.
- calculate reflection matrix of right substrate plus subvolume n.
- use this to calculate right embedding potential on S_{n-1}.
- calculate reflection matrix of right substrate plus subvolumes $n-1$ and n.
- use this to calculate right embedding potential on S_{n-2}.
- repeat until all right embedding potentials are found.

The second approach uses the fact that (3.55) is valid for any waves on the right and left of subvolume n, so that we get:

$$\begin{pmatrix} \Psi_{n-1} \\ \Psi'_{n-1} \end{pmatrix} = \mathbb{T}_n^{\leftarrow} \times \begin{pmatrix} \Psi_n \\ \Psi'_n \end{pmatrix} \tag{3.110}$$

where the prime indicates a normal derivative with the appropriate sign and suffix $n-1$ is to the left of subvolume n and suffix n is to the right. From (3.110) we get the following two equations:

$$\Psi_{n-1} = \mathbb{T}_{11}^{\leftarrow}\Psi_n + \mathbb{T}_{12}^{\leftarrow}\Psi'_n \tag{3.111a}$$
$$\Psi'_{n-1} = \mathbb{T}_{21}^{\leftarrow}\Psi_n + \mathbb{T}_{22}^{\leftarrow}\Psi'_n \tag{3.111b}$$

But we have also defined the embedding potential as a logarithmic derivative. This leads to:

$$\Psi'_n = 2\Sigma_n\Psi_n \tag{3.112a}$$
$$\Psi'_{n-1} = 2\Sigma_{n-1}\Psi_{n-1} \tag{3.112b}$$

Substituting (3.112b) into (3.111) we obtain:

$$\Psi_{n-1} = (\mathbb{T}_{11}^{\leftarrow} + 2\mathbb{T}_{12}^{\leftarrow}\Sigma_n)\,\Psi_n \tag{3.113a}$$
$$\Psi'_{n-1} = (\mathbb{T}_{21}^{\leftarrow} + 2\mathbb{T}_{22}^{\leftarrow}\Sigma_n)\,\Psi_n \tag{3.113b}$$

Inverting (3.113a) and substituting into (3.113b) gives us:

$$\Psi'_{n-1} = (\mathbb{T}_{21}^{\leftarrow} + 2\mathbb{T}_{22}^{\leftarrow}\Sigma_n)\,(\mathbb{T}_{11}^{\leftarrow} + 2\mathbb{T}_{12}^{\leftarrow}\Sigma_n)^{-1}\,\Psi_{n-1} \tag{3.114}$$

Using (3.112b) we see that:

$$2\Sigma_{n-1} = (\mathbb{T}_{21}^{\leftarrow} + 2\mathbb{T}_{22}^{\leftarrow}\Sigma_n)\,(\mathbb{T}_{11}^{\leftarrow} + 2\mathbb{T}_{12}^{\leftarrow}\Sigma_n)^{-1} \tag{3.115}$$

This relates the embedding potential Σ_{n-1} directly to the embedding potential Σ_n through the transfer matrix. We can use this to propagate embedding potential through subvolumes, using the transfer matrices for left-to-right transfer for the left-hand embedding potentials and the right-to-left transfer matrices for the right-hand embedding potentials.
In practice it turns out that we can also use (3.115) to find embedding potentials for semi-infinite substrates, but this is sometimes unstable and does not converge to the physical solution: we prefer to use the procedure given in sections 3.4.2.4 and 3.4.2.5.
In the remainder of this chapter I want to outline the plan of attack for the different systems that the embedding method for layered systems can handle.

3.5 A bulk system embedded

The simplest system we can embed is a bulk system. Of course there are lots of other methods which can handle a periodic bulk, but we want to do it using embedding because this enables us to compare surface or interface with bulk calculated with the same method. In figure 3.13 I show a flow chart of the way the bulk is calculated.

1. First we read in the geometry data of the system, i.e. how many subvolumes, how many atoms in each subvolume, the atomic positions, etc. Then we read in the potentials for all subvolumes.

2. With all this input we can then construct the Hamiltonian in each subvolume and construct the Green function with zero normal derivative on both sides. From this Green function we can construct the scattering properties, according to (3.24) - (3.44).

3. Using the scattering matrices and transfer matrices, we construct the transfer matrix for the *bulk repeat unit* (BRU): this is the unit which is periodic in the direction perpendicular to the subvolumes. The BRU can be built up out of one or more subvolumes.

4. Then using the transfer-matrix for the BRU, we can construct the left substrate by stacking BRU's until we reach convergence, which gives us Σ_{∞}^{L}. We then also construct the right substrate, which gives us Σ_{∞}^{R}.

5. These two substrate embedding potentials, together with the subvolume scattering properties calculated earlier, give us all left and right embedding potentials for the BRU.

6. Using these we can now calculate the Green functions (this time with proper boundary conditions) for all subvolumes in the BRU.

7. From these Green functions we calculate the physical properties we are interested in, for example a charge density $\rho(\mathbf{r})$ or a density of states for a single $\mathbf{k}_{\|}$, $\sigma_{\mathbf{k}_{\|}}(E)$.

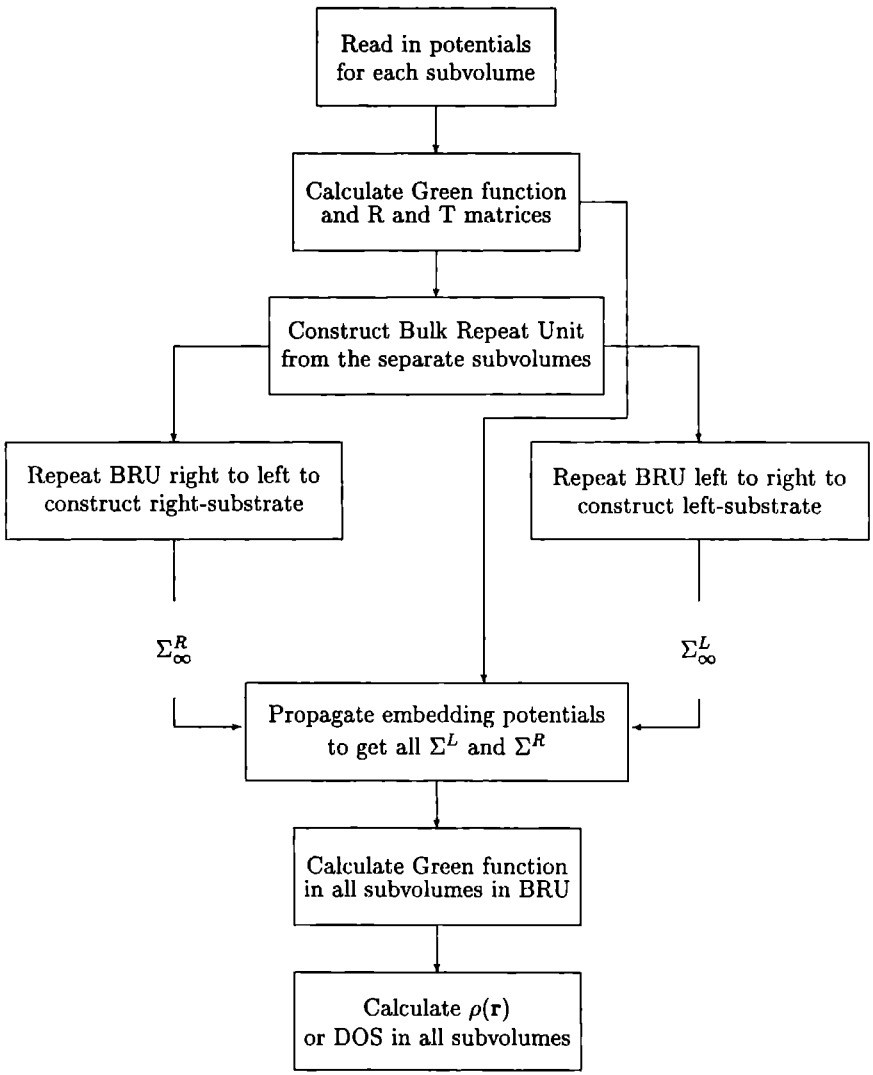


Figure 3.13: *The embedding approach applied to a bulk system.*

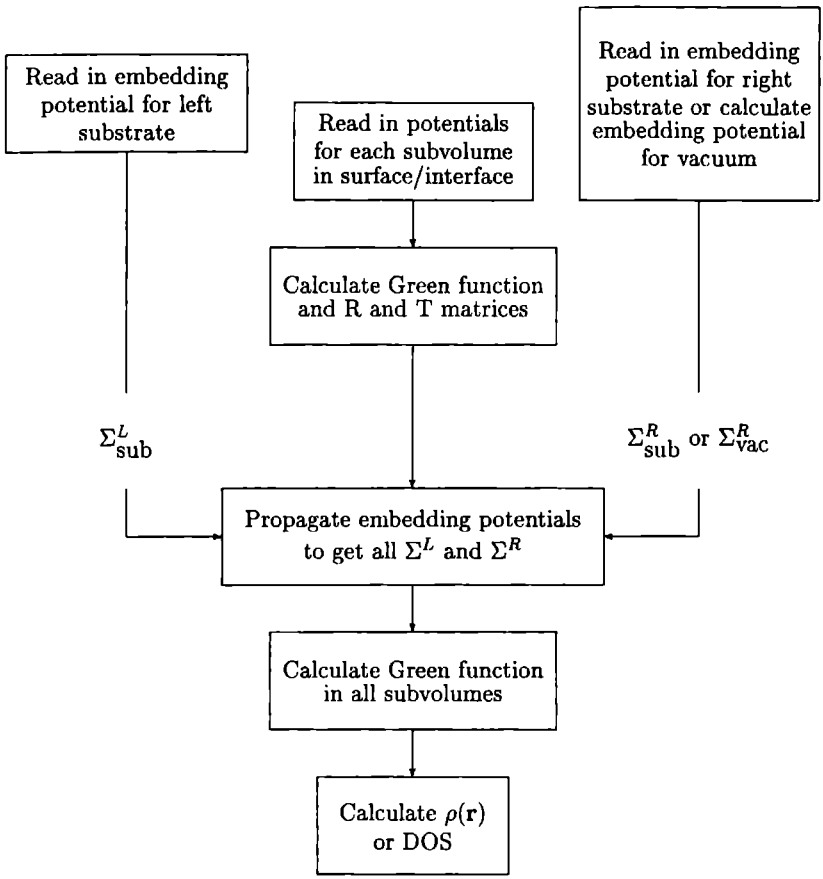


Figure 3.14: *The embedding approach applied to an interface/surface system.*

3.6 Surfaces and interfaces embedded

To embed interfaces and surfaces we first calculate appropriate left and right substrate embedding potentials. For an interface these correspond to different semi-infinite bulk materials; for the surface one is for a bulk material, the other is for the vacuum. In between we have a stack of subvolumes - not periodic in the z-direction, unlike the bulk. The flow of steps showing how a surface or interface calculation progresses is shown in figure 3.14. When we break the flow chart down into steps defined horizontally, we get:

1. We start by reading in the geometry, potentials for the stack of subvolumes in the interface/surface, etc. We also read in the embedding potential for the left substrate. And in the case of an interface for the right substrate. For a surface however, we always take the vacuum to be on the right, so we have to calculate the vacuum embedding

potential (this can be done analytically). The vacuum embedding potential is given by:

$$\Sigma^{\text{vac},\mathbf{k}_\parallel}_{\mathbf{G}\mathbf{G}'}(E) = -\tfrac{1}{2}\delta_{\mathbf{G}\mathbf{G}'}\sqrt{|\mathbf{G}+\mathbf{k}_\parallel|^2 - 2E} \tag{3.116}$$

2. From the potentials and geometry we calculate the Hamiltonian, which gives us the Green function. And from the Green function the scattering matrices can be calculated.

3. Using the read in or calculated embedding potentials on the sides of the stack of subvolumes we can calculate the embedding potentials on both sides of all subvolumes.

4. We can now calculate the Green function with the correct boundary conditions.

5. These Green functions give us the physical quantities, as in the bulk case.

In this chapter I have outlined some of the derivations forming the foundations of any code based on the embedding approach. It is not complete as some of the details have not been included, but including all details would lead to an enormous amount of formulae. Some of the details will be mentioned in the chapters on implementation. In the following chapter I will look at a nice application of the scattering matrices we have introduced in this chapter.

Chapter 4

Embedding and electron transport

In this chapter I shall make a link between the embedding method and transport. Actually the only part of embedding we need is the part for finding the scattering properties of subvolumes. Those properties are what we shall use to calculate the conductance. The starting point for our calculation of the conductance is the Landauer-Büttiker formalism. I shall briefly discuss this formalism and then make the link to the details of the embedding method described in chapter 3.

4.1 Landauer-Büttiker formalism

We will limit our discussion of conductance to the ballistic regime. In ballistic transport it is the scattering of the conduction electrons at the sample boundaries that limits the current, and not impurity scattering. The canonical example of a ballistic conductor is the point contact, as shown in figure 4.1. When a voltage difference V is applied between the wide regions to the left and right (the leads), the current I through the small constriction is *finite*, even in the complete absence of impurities. This current is finite because of scattering of the electrons at the entrance to the constriction.
The *contact conductance* is defined as:

$$G = \frac{I}{V} \tag{4.1}$$

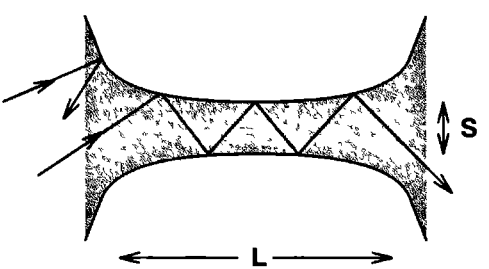


Figure 4.1: *Schematic picture of a point contact (shaded area). All scattering is specular. In general $L \ll l$, where l is the mean free path of the electrons.*

This quantity is proportional to the area of the point contact S, but not to its length L, as long as $L \ll l$, where l is the mean free path of the electrons. The fact that the conductance of the constriction is independent of its length should make it clear that we cannot use a local conductivity argument to find the conductance, as we can in the diffusive transport regime.

We shall use the Landauer formula[8, 9] to link the conductance G to a Fermi level property of the sample, namely the transmission probability of propagating modes at the Fermi energy (so-called *quantum channels*). These propagating modes do not scatter in the leads and each carries unit flux (i.e. one electron per channel).
The Landauer formula can be written as:

$$G = \frac{2e^2}{h} \sum_{n,m=1}^{N} |t_{mn}|^2 \tag{4.2}$$

The factor 2 comes from the spin-degeneracy, because at this stage we only consider non-magnetic conductors. The conductance is now expressed in terms of the transmission matrix $\mathbf{t}$. The matrix element t_{mn} is the transmission probability amplitude of state n on the left into state m on the right. So, to find the conductance for a point contact in the ballistic regime, we have to know how these conducting states propagate through the constriction.

In this thesis we concentrate on calculating the conductance of interfaces. This means that we retain two-dimensional periodicity parallel to the interface, corresponding to a point contact of infinite area S. If we use periodic boundary conditions parallel to the interface, quantizing over area S, the sums over channels in (4.2) become sums over parallel wavevectors $\mathbf{k}_\parallel$, $\mathbf{k'}_\parallel$ and bands μ, ν at the Fermi energy:

$$G = \frac{2e^2}{h} \sum_{\mathbf{k}_\parallel,\mu} \sum_{\mathbf{k'}_\parallel,\nu} |t_{\mathbf{k'}_\parallel \mathbf{k}_\parallel,\nu\mu}|^2 \tag{4.3}$$

At each $\mathbf{k}_\parallel$ μ runs over the propagating states at the Fermi level for that wave-vector, similarly for ν at $\mathbf{k'}_\parallel$. Since there is an infinite number of these wave-vectors (for the point-contact area going to infinity), we should actually integrate over this variable, obtaining:

$$G = \frac{2e^2}{h} \frac{S}{4\pi^2} \int \mathrm{d}\mathbf{k}_\parallel \int \mathrm{d}\mathbf{k'}_\parallel \sum_{\mu,\nu} |t_{\mathbf{k'}_\parallel \mathbf{k}_\parallel,\nu\mu}|^2 \tag{4.4}$$

where the factor $\frac{S}{4\pi^2}$ is introduced because for any large but finite S we have a particular density of wave-vectors in (4.4). However, eventually, we replace the integral by a sum over wave-vectors within the first Brillouin zone. This leads to the following expression:

$$G = \frac{2e^2}{h} \frac{S}{4\pi^2} \frac{4\pi^2}{A} \sum_{\mathbf{k}_\parallel,\mu} \sum_{\mathbf{k'}_\parallel,\nu} |t_{\mathbf{k'}_\parallel \mathbf{k}_\parallel,\nu\mu}|^2 \tag{4.5}$$

Going from the integral to the sum leads to the factor $\frac{4\pi^2}{A}$, where A is the area of the interface in the unit cell. $t_{\mathbf{k'}_\parallel \mathbf{k}_\parallel,\nu\mu}$ is now the transmission probability amplitude for state μ with wave-vector $\mathbf{k}_\parallel$ to the left of the interface into state ν with wave-vector $\mathbf{k'}_\parallel$ to the right of the interface. Eventually we are interested in the conductance per unit area, and we write:

$$\frac{G}{S} = \frac{2e^2}{Ah} \sum_{\mathbf{k}_\parallel,\mu} \sum_{\mathbf{k'}_\parallel,\nu} |t_{\mathbf{k'}_\parallel \mathbf{k}_\parallel,\nu\mu}|^2 \tag{4.6}$$

(4.6) is the formula I will use in the remainder of this thesis when I talk about the Landauer equation.
In the next section we shall examine (4.6) in detail and examine whether we can gather the ingredients from the embedding method.

4.2 Can we find all the ingredients?

In the ballistic regime, we can safely assume that the conduction electrons only scatter at the interface, travelling through the bulk of both leads without scattering. I shall now show that the electrons we are looking for are the propagating Bloch electrons.

4.2.1 Bloch electrons

Bloch electrons are electrons in the states in the three-dimensional crystal which do not scatter off the lattice, and are able to propagate through the bulk indefinitely. The Bloch states $b_{\mathbf{k}}(\mathbf{r})$ obey Bloch's theorem:

$$b_{\mathbf{k}}(\mathbf{r}+\mathbf{r}_t) = e^{i\mathbf{k}\,\mathbf{r}_t} b_{\mathbf{k}}(\mathbf{r}) \tag{4.7}$$

where $\mathbf{r}_t$ is any lattice vector. To relate this to the transfer matrix, let us choose $\mathbf{r}_t$ to be the vector that takes us from the left-origin of the bulk repeat unit (BRU) to the right-origin. Then we know:

$$b_{\mathbf{k}}(\mathbf{r}+\mathbf{r}_t) = \mathbb{T}_{\mathrm{BRU}} b_{\mathbf{k}}(\mathbf{r}) \tag{4.8}$$

where $\mathbb{T}_{\mathrm{BRU}}$ is the left-to-right transfer matrix of the BRU. Combining (4.7) and (4.8) we obtain:

$$\mathbb{T}_{\mathrm{BRU}} b_{\mathbf{k}}(\mathbf{r}) = e^{i\mathbf{k}\cdot\mathbf{r}_t} b_{\mathbf{k}}(\mathbf{r}) \tag{4.9}$$

We have now found a way to calculate the Bloch states, because they are the eigenstates of the transfer matrix with eigenvalues $e^{i\mathbf{k}\,\mathbf{r}_t}$. We shall see in "Part II: Implementation" that (4.9) is used in the computer code to calculate the band-structure of a bulk material.

4.2.2 Finding the transmission of Bloch electrons

In this section we shall find the transmission of the Bloch states through a subvolume or stack of subvolumes. We know from (3.60) and (3.62) that we can get the transmission matrix from the transfer matrix. However, all the transfer matrices which we have calculated up to now are in terms of our basis functions given by (3.12) and (3.39). Can we write the transfer matrix of a subvolume in terms of Bloch states of the corresponding bulk? The answer is yes – via a basis transformation. This is not a trivial transformation, because the basis we use in the inter-subvolume region ((3.12), (3.39)) is not orthogonal. We start with an expansion of a Bloch state in terms of the inter-subvolume basis:

$$b_{\mathbf{k}}(\mathbf{r}) = \sum_{l=1}^{2n} \alpha_{\mathbf{k},l}\,\phi_l(\mathbf{r}) \tag{4.10}$$

where n is the size of the basis, i.e. the dimension of the scattering matrices, and l indicates the basis functions. The ϕ_l's are given by:

$$\phi_l(\mathbf{r}) = \begin{cases} \frac{1}{\sqrt{A}} e^{\gamma_{\mathbf{G}}^{+} z} e^{i(\mathbf{k}_{\parallel}+\mathbf{G})\cdot\mathbf{R}} & \text{if } l \leq n \\ \frac{1}{\sqrt{A}} e^{\gamma_{\mathbf{G}}^{-} z} e^{i(\mathbf{k}_{\parallel}+\mathbf{G})\cdot\mathbf{R}} & \text{if } n < l \leq 2n \end{cases} \tag{4.11}$$

where l runs over the different $\mathbf{G}$'s twice. In general we have $2n$ Bloch states, n of them moving in the $+z$ direction, and n moving in the $-z$ direction.
We need a transfer matrix in terms of Bloch states, i.e. we want a transformation from the Bloch states to the basis functions. Using some linear algebra, we can show that the transformation matrix will look like:

$$\mathbb{U}_{l,\mathbf{k}} = \alpha_{\mathbf{k},l} \tag{4.12}$$

The inverse transformation, is the inverse of the matrix $\mathbb{U}$. This means that the transfer matrix in terms of Bloch states is written as:

$$\mathbb{T}_{\text{Bloch}} = \mathbb{U}^{-1}\mathbb{T}\mathbb{U} \tag{4.13}$$

We have seen that from the expansion of the eigenvectors of $\mathbb{T}$ we can get the transformation matrix $\mathbb{U}$, which enables us to construct the transfer matrix for the Bloch states. From this transfer matrix we can get the transmission matrix, using (3.60) and (3.62). However as we shall see in the next section we do not need all elements of this transmission matrix.

4.2.3 Not all Bloch electrons carry current

Since the transfer matrix is $2n \times 2n$, we will find $2n$ eigenstates and also $2n$ eigenvalues. But we know that not all materials have $2n$ states at the Fermi level. To find an answer to this problem, let us examine the eigenvalues, which should give us the band-structure. The eigenvalues of the transfer matrix are of the form:

$$\mu_l = e^{i\mathbf{k}_l \cdot \mathbf{r}_t} \tag{4.14}$$

This means that we can find the $\mathbf{k}_l$ values from the eigenvalues using:

$$\mathbf{k}_l \cdot \mathbf{r}_t = \frac{\ln(\mu_l)}{i} \tag{4.15}$$

Since we know $\mathbf{r}_t$ and the parallel part of $\mathbf{k}_l$ is fixed in our method, we can write:

$$(\mathbf{k}_l)_z = \left[\frac{\ln(\mu_l)}{i} - (\mathbf{k}_l)_x(\mathbf{r}_t)_x - (\mathbf{k}_l)_y(\mathbf{r}_t)_y\right] \frac{1}{(\mathbf{r}_t)_z} \tag{4.16}$$

These $(\mathbf{k}_l)_z$ need not all be real. The propagating states, able to travel through a macroscopic part of the crystal in the z direction, all have purely real $(\mathbf{k}_l)_z$. This is because the imaginary part of the $(\mathbf{k}_l)_z$ will lead to the state decaying exponentially with increasing or decreasing z.
This means that for our conductance problem we only need the transmission probability of these propagating states, i.e. the Bloch states with purely real $(\mathbf{k}_l)_z$. This limits the amount of data we need to store to be able to calculate the conductance. Another reduction in complexity is the fact that since we only look at scattering at an interface periodic in two

dimensions, the parallel momentum is conserved, since we do not have any impurity scattering which might change the parallel momentum of the conduction electron. This means that we have the same $\mathbf{k}_{\|}$ on both sides of the interface or alternatively that t is diagonal in $\mathbf{k}_{\|}$, i.e. (4.6) becomes:

$$\frac{G}{S} = \frac{2e^2}{Ah} \sum_{\mathbf{k}_{\|},\nu\mu} |t_{\mathbf{k}_{\|}\mathbf{k}_{\|},\nu\mu}|^2 \tag{4.17}$$

We now have all ingredients that we need to calculate the conductance.

4.3 Conductance of a bulk system

We will now discuss how to calculate the conductance in a typical bulk system. In this case the transfer matrix in terms of the Bloch states will be a diagonal matrix, because the Bloch states are the eigenstates of the transfer matrix. Also, the diagonal elements are the eigenvalues. Since we only take into account the propagating states, this means that $|t_{\mathbf{k}_{\|}\mathbf{k}_{\|},\nu\nu}|^2$ is unity for all the states carrying current.

So for a bulk system, all that remains for calculating the conductance is to count the states at the Fermi level, for all $\mathbf{k}_{\|}$. This means the conductance is proportional to the surface area of the projected Fermi surface in $\mathbf{k}_{\|}$-space.

4.4 Conductance of an interface

In this section we will study an interface between material A and material B, a case which is more complicated that the bulk. A schematic picture of the interface is shown in figure 4.2. The interface has to be commensurate, i.e. both materials share a common periodicity in the parallel direction. We need only consider the two subvolumes in the middle, one of material A and one of material B. Before we concentrate on the interface, we must calculate the bulk of both materials. For each $\mathbf{k}_{\|}$ we need the transformation matrix $\mathbb{U}$, to be able to transfer between Bloch states and our basis, and the number of propagating Bloch states. The Bloch states in the transformation matrix are arranged such that the propagating ones are always number 1 to M, where M is the number of propagating states.

We consider the current flowing from left to right. The calculation of the conductance of the interface then goes along the following lines:

- Pick a $\mathbf{k}_{\|}$
- Calculate the transfer matrix for each subvolume, and from these the transfer matrix for the stack of the two subvolumes which form the interface.

$$\mathbb{T}^{\rightarrow}_{AB} = \mathbb{T}^{\rightarrow}_{B} \times \mathbb{T}^{\rightarrow}_{A} \tag{4.18}$$

- This transfer matrix is in our normal basis. We transform it to the Bloch states basis, using the right-hand Bloch states on the right and the left-hand Bloch states on the left.

$$\mathbb{T}^{\rightarrow}_{\text{Bloch}} = \mathbb{U}_{B}^{-1} \mathbb{T}^{\rightarrow}_{AB} \mathbb{U}_{A} \tag{4.19}$$

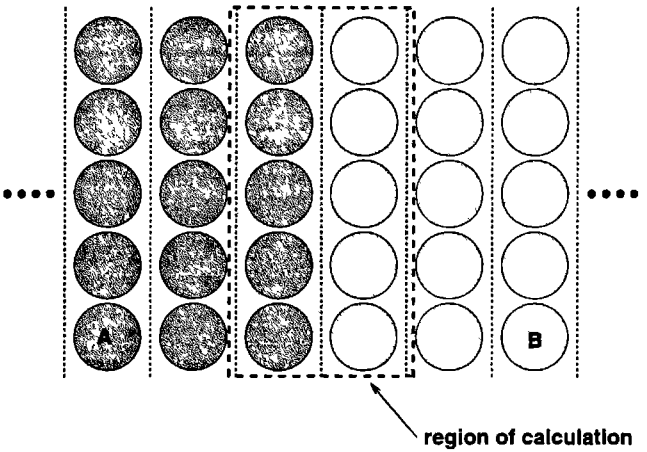


Figure 4.2: *Schematic picture of interface between materials A and B, both extending to infinity on left and right. The shaded region in the center containing two subvolumes will be where we will be performing our calculation.*

- From this $\mathbb{T}^{\rightarrow}_{\text{Bloch}}$ we can now construct the transmission matrix $\mathbf{T}$ using the expression for the transfer matrix given in (3.62).
- We now know that:

$$t_{\mathbf{k}_{\|}\mathbf{k}_{\|},\nu\mu} = T_{\nu\mu} \tag{4.20}$$

where μ runs only over the propagating Bloch states to the left and ν only over the propagating Bloch states to the right. $t_{\mathbf{k}_{\|}\mathbf{k}_{\|},\nu\mu}$ is the transmission probability amplitude between propagating Bloch state μ in A, incident on the interface from the left, and propagating Bloch state ν on the right.

All that remains is to sum over $\mathbf{k}_{\|}$, put the pre-factor $\frac{2e^2}{Ah}$ in front and we have calculated $\frac{G}{S}$ for the A/B interface.

4.5 Is it really that simple?

The process described in the previous section seems rather simple. Of course it is not as simple as that, but the basic outline is indeed simple. There are some small details that are non-trivial, and one of these will now be discussed.

4.5.1 Flux normalisation

In the Landauer formalism, as expressed in (4.6), it is assumed that the states $(\mathbf{k}_{\|}, \nu)$ and $(\mathbf{k}'_{\|}, \mu)$ are normalised to carry unit current. In other words, each channel can at most contain one electron (two in the case of a non-magnetic material). However the basis we have defined in chapter 3 in the inter-subvolume-region is not normalised. Thus the Bloch states we get by

diagonalising the transfer matrix are not normalised. This means that the $t_{\mathbf{k}_\parallel \mathbf{k}_\parallel, \nu\mu}$ as found in the previous section have to be corrected for this normalisation.

We can perform this correction in the following way. In the bulk calculation where we find U and M for each $\mathbf{k}_\parallel$, we also calculate the amount of flux in each propagating Bloch state. To calculate the flux of current in a Bloch state in the z direction, we use the definition of current:

$$I_{k_z} = \frac{i}{2} \int \mathrm{d}^2\mathbf{r} \left[b_\mathbf{k}(\mathbf{r}) \frac{\partial}{\partial z} b^*_\mathbf{k}(\mathbf{r}) - b^*_\mathbf{k}(\mathbf{r}) \frac{\partial}{\partial z} b_\mathbf{k}(\mathbf{r}) \right] \tag{4.21}$$

where $b_\mathbf{k}(\mathbf{r})$ is the Bloch state for which we want to calculate the current-flux. $\mathbf{k}$ is always a wave-vector for which $\mathbf{k}_\parallel$ is fixed and k_z indicates the state, thus we only label the current with k_z. The integration in (4.21) is over the surface of the unit cell, across which we want to know the flux. We now write $b_\mathbf{k}(\mathbf{r})$ in terms of left and right traveling basis functions:

$$b_\mathbf{k}(\mathbf{r}) = \sum_\mathbf{G} \left[A_{\mathbf{G},k_z} e^{i(\mathbf{k}_\parallel + \mathbf{G})\,\mathbf{R}} e^{\gamma^+_\mathbf{G} z} + B_{\mathbf{G},k_z} e^{i(\mathbf{k}_\parallel + \mathbf{G})\,\mathbf{R}} e^{\gamma^-_\mathbf{G} z} \right] \tag{4.22}$$

where the factor $\frac{1}{\sqrt{A}}$ is moved into the A's and B's, and $\gamma^\pm_\mathbf{G}$ is defined as in (3.13), namely:

$$\gamma^\pm_\mathbf{G} = \mp \sqrt{|\mathbf{k}_\parallel + \mathbf{G}|^2 - 2E} \tag{4.23}$$

Using (4.22) we can write:

$$\begin{aligned} b_\mathbf{k}(\mathbf{r}) \frac{\partial}{\partial z} b^*_\mathbf{k}(\mathbf{r}) = \sum_\mathbf{G} \sum_{\mathbf{G}'} & \left[A_{\mathbf{G},k_z} e^{i(\mathbf{k}_\parallel + \mathbf{G})\,\mathbf{R}} e^{\gamma^+_\mathbf{G} z} + B_{\mathbf{G},k_z} e^{i(\mathbf{k}_\parallel + \mathbf{G})\,\mathbf{R}} e^{\gamma^-_\mathbf{G} z} \right] \times \\ & \left[{\gamma^+_{\mathbf{G}'}}^* A^*_{\mathbf{G}',k_z} e^{-i(\mathbf{k}_\parallel + \mathbf{G}')\,\mathbf{R}} e^{{\gamma^+_{\mathbf{G}'}}^* z} + {\gamma^-_{\mathbf{G}'}}^* B^*_{\mathbf{G}',k_z} e^{-i(\mathbf{k}_\parallel + \mathbf{G}')\,\mathbf{R}} e^{{\gamma^-_{\mathbf{G}'}}^* z} \right] \end{aligned} \tag{4.24}$$

Integrated over the surface of the unit cell this becomes:

$$\begin{aligned} \int \mathrm{d}^2\mathbf{r} b_\mathbf{k}(\mathbf{r}) \frac{\partial}{\partial z} b^*_\mathbf{k}(\mathbf{r}) &= \sum_\mathbf{G} \left[A_{\mathbf{G},k_z} e^{\gamma^+_\mathbf{G} z} + B_{\mathbf{G},k_z} e^{\gamma^-_\mathbf{G} z} \right] \times \\ &\quad \left[{\gamma^+_\mathbf{G}}^* A^*_{\mathbf{G},k_z} e^{{\gamma^+_\mathbf{G}}^* z} + {\gamma^-_\mathbf{G}}^* B^*_{\mathbf{G},k_z} e^{{\gamma^-_\mathbf{G}}^* z} \right] \\ &= \sum_\mathbf{G} \Big[{\gamma^+_\mathbf{G}}^* A_{\mathbf{G},k_z} A^*_{\mathbf{G},k_z} e^{(\gamma^+_\mathbf{G} + {\gamma^+_\mathbf{G}}^*) z} + {\gamma^-_\mathbf{G}}^* A_{\mathbf{G},k_z} B^*_{\mathbf{G},k_z} e^{(\gamma^+_\mathbf{G} + {\gamma^-_\mathbf{G}}^*) z} + \\ &\qquad {\gamma^+_\mathbf{G}}^* B_{\mathbf{G},k_z} A^*_{\mathbf{G},k_z} e^{(\gamma^-_\mathbf{G} + {\gamma^+_\mathbf{G}}^*) z} + {\gamma^-_\mathbf{G}}^* B_{\mathbf{G},k_z} B^*_{\mathbf{G},k_z} e^{(\gamma^-_\mathbf{G} + {\gamma^-_\mathbf{G}}^*) z} \Big] \end{aligned} \tag{4.25}$$

The second term in (4.21) is the complex conjugate of (4.25). This means that we get, using (4.21), (4.25) and its complex conjugate:

$$\begin{aligned} I_{k_z} = \frac{i}{2} \sum_\mathbf{G} \Big[& \left({\gamma^+_\mathbf{G}}^* - \gamma^+_\mathbf{G} \right) A_{\mathbf{G},k_z} A^*_{\mathbf{G},k_z} e^{(\gamma^+_\mathbf{G} + {\gamma^+_\mathbf{G}}^*) z} \\ & + \left({\gamma^+_\mathbf{G}}^* + \gamma^+_\mathbf{G} \right) A^*_{\mathbf{G},k_z} B_{\mathbf{G},k_z} e^{(\gamma^-_\mathbf{G} + {\gamma^+_\mathbf{G}}^*) z} \\ & - \left({\gamma^+_\mathbf{G}}^* + \gamma^+_\mathbf{G} \right) A_{\mathbf{G},k_z} B^*_{\mathbf{G},k_z} e^{(\gamma^+_\mathbf{G} + {\gamma^-_\mathbf{G}}^*) z} \\ & - \left({\gamma^+_\mathbf{G}}^* - \gamma^+_\mathbf{G} \right) B_{\mathbf{G},k_z} B^*_{\mathbf{G},k_z} e^{(\gamma^-_\mathbf{G} + {\gamma^-_\mathbf{G}}^*) z} \Big] \end{aligned} \tag{4.26}$$

Here we have used properties like $\gamma_{\mathbf{G}}^{+} = -\gamma_{\mathbf{G}}^{-}$.

Since we can evaluate the flux for a Bloch state at any z (the state doesn't scatter and therefore retains all its flux), we choose $z = 0$, which simplifies (4.26) to:

$$I_{k_z} = \frac{i}{2} \sum_{\mathbf{G}} \left[\left(\gamma_{\mathbf{G}}^{+*} - \gamma_{\mathbf{G}}^{+} \right) A_{\mathbf{G},k_z} A_{\mathbf{G},k_z}^{*} + \left(\gamma_{\mathbf{G}}^{+*} + \gamma_{\mathbf{G}}^{+} \right) A_{\mathbf{G},k_z}^{*} B_{\mathbf{G},k_z} \right.$$
$$\left. - \left(\gamma_{\mathbf{G}}^{+*} + \gamma_{\mathbf{G}}^{+} \right) A_{\mathbf{G},k_z} B_{\mathbf{G},k_z}^{*} - \left(\gamma_{\mathbf{G}}^{+*} - \gamma_{\mathbf{G}}^{+} \right) B_{\mathbf{G},k_z} B_{\mathbf{G},k_z}^{*} \right] \quad (4.27)$$

In this way we can calculate the flux in all the channels that contribute to the conductance.

We use these fluxes of the Bloch states to correct the transmission properties of the Bloch states before they go into the Landauer formula. So instead of (4.20) we use:

$$t_{\mathbf{k}_{\|}\mathbf{k}_{\|},\nu\mu} = T_{\nu\mu} \sqrt{\frac{I_{(k_z)_\nu}}{I_{(k_z)_\mu}}} \quad (4.28)$$

In words, we multiply the transmission with the flux in the right Bloch state ν and divide by the flux in the left Bloch state μ.

4.6 Which interfaces can we handle?

What are the limitations to the interfaces which can be handled using the methods described in this chapter? Actually the limitations are set by the methods of the previous chapter. Any bulk that can be handled using embedding techniques can be used to obtain Bloch states and basis transformation matrices $\mathbb{U}$. The same is true for interface configurations. If the interface can be broken down into subvolumes, and can be handled by the embedding method (i.e. is commensurate), we can use the techniques of this chapter to calculate the conductance. In a later chapter we will describe an extra method to be able to handle magnetic domain walls of a finite thickness.

Part II

Implementation

Chapter 5

From method to computer program

In this chapter I shall lay the foundations for the rest of this part of the thesis about implementation. By implementation I do not necessarily mean only putting the method into a computer code, but also the other choices which have to be made about the method, such as basis sets and which geometries to handle. Also in some cases I will comment on a specific way to code something up, if that results in greatly increasing efficiency of the resulting code. Whenever a subroutine is mentioned it will be written in typeface, for example `mtme_lapw` is the subroutine which calculates the muffin-tin matrix elements for a LAPW basis.

5.1 Breaking the method up into pieces

Any electronic structure code which uses density functional theory can logically be broken up into different parts. One of the results of DFT is that we always have to iterate to self-consistency. We can choose the variable we want to iterate on: either potential or charge-density. To be able to do either, I have decided to make the computer code modular, using three building blocks, which can then be executed in a certain order. In the following section I will briefly introduce each module. Each of these modules is a separate computer program.

5.2 GROWing charge densities

The first code, called `GROW` solves the Schrödinger equation in the region of interest, using all the embedding techniques described in chapter 3. As input it takes a potential and geometry information. Dependent on which mode this code is run in, it can generate the outputs as shown in table 5.1. The `GROW` code is actually that part of the total computer code which

mode	output
-1	band structure
0	different projections of densities of states
1	charge density with fixed Fermi energy
2	charge neutral charge density (fermi-search)

Table 5.1: *Different outputs of the* `GROW` *code.*

contains the embedding method That is why some modes of this code will input or output embedding potentials or scattering properties This also means that the GROW code will be the module which will be adapted to calculate conductances This adapted module will be discussed in chapter 9 The 'normal' GROW module will be discussed in detail in chapter 6

5.3 FISHing for potentials

The second module is called FISH, as it solves *Poisson's* equation It takes as input a charge density and geometry information and outputs a potential To solve Poisson's equation, this code uses Weinert's method [10] to solve the electrostatic potential problem, extended to cope with interface boundary conditions across subvolumes This module is also the part of the program which incorporates the exchange-correlation potential as discussed in section 1 2 3 in chapter 1 The module can actually handle several schemes for calculating the exchange-correlation potential The scheme chosen depends on one input variable in the global control file The FISH module will be discussed in greater detail in chapter 7

5.4 MIXing it all up

The MIX module is the last part of the code It enables us to perform the self-consistency iteration loop MIX has two different I/O-modes, one for potentials and one for charge-densities The MIX module inputs two or more potentials/charge-densities and outputs a potential/charge-density which is the mixed result of the inputs There are different mixing modes, one being straightforward simple mixing, but we can use more elaborate mixing schemes to ensure faster convergence The different ways to converge to self-consistency are show in figure 5 1 for potentials, and figure 5 2 for charge densities The details of the MIX module are discussed in chapter 8

The remainder of Part II *Implementation* will go into the details of the different modules mentioned in this chapter

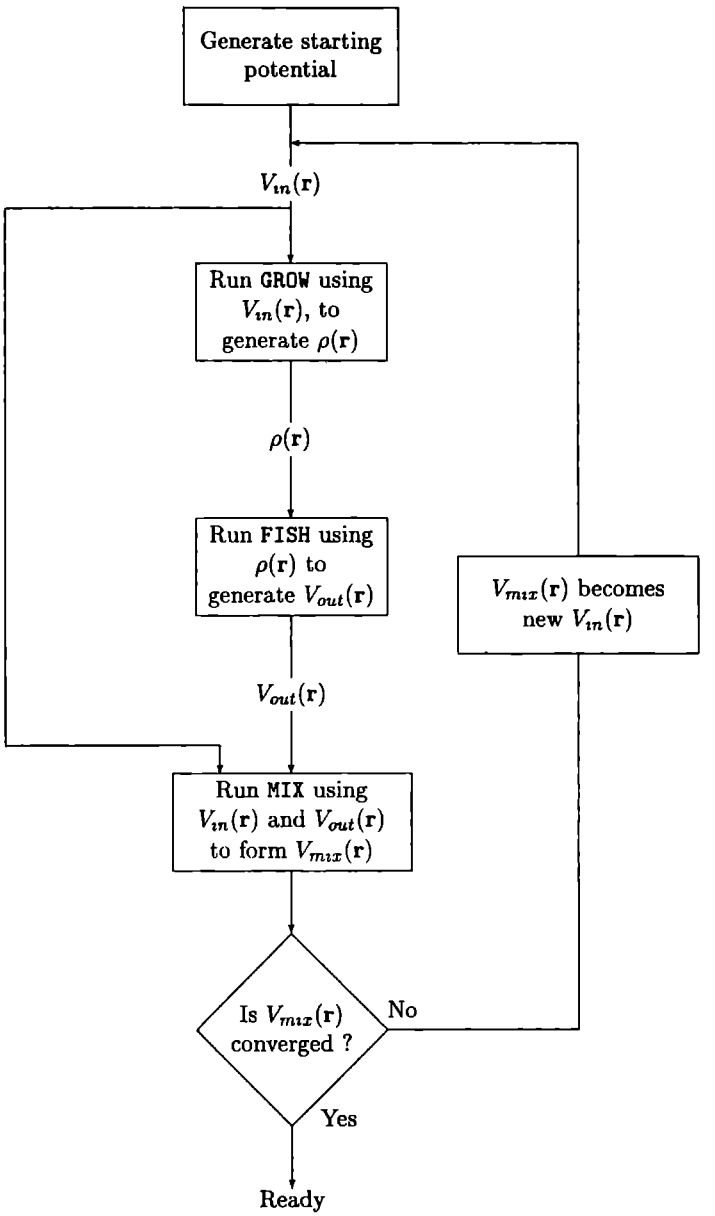


Figure 5.1: *Flow chart showing how the different modules can be linked using* MIX *to converge to self-consistency, with the potential as an iteration variable.*

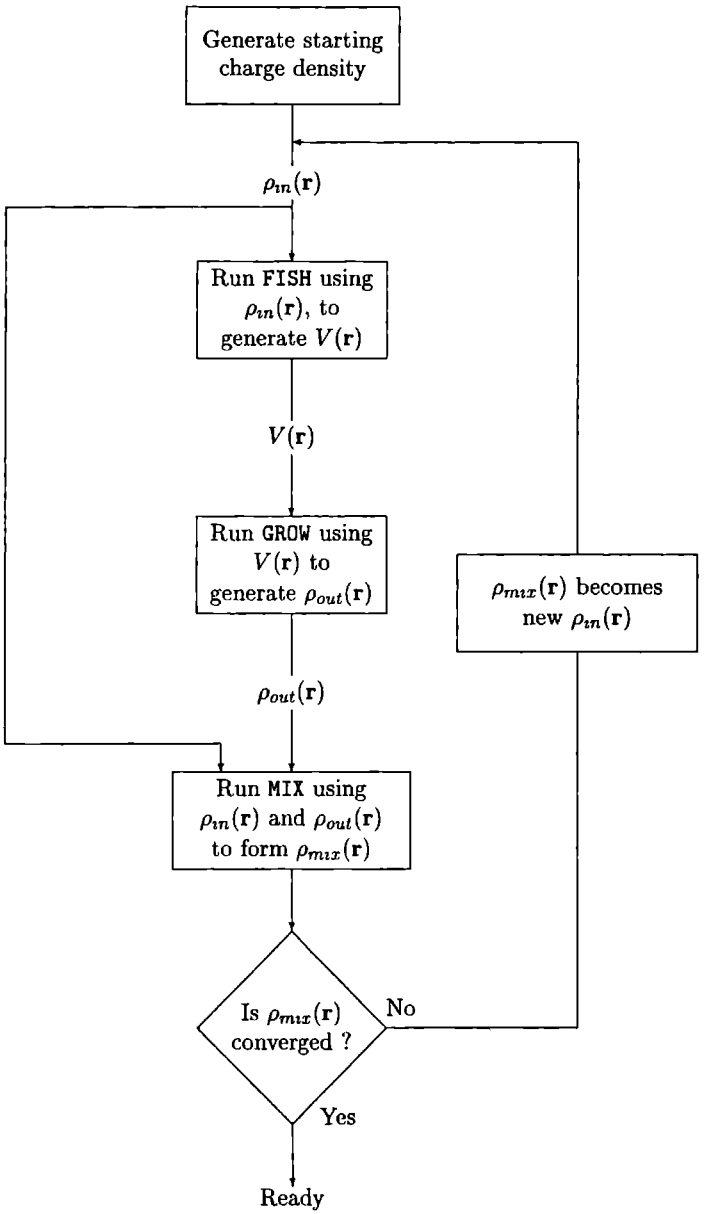


Figure 5.2: *Flow chart showing how the different modules can be linked using* `MIX` *to converge to self-consistency, with the charge-density as an iteration variable.*

Chapter 6

How to GROW charge densities

This chapter is dedicated to the module GROW of the embedding code. We discuss some of the details of this module, but of course it is impossible to include everything, since the actual module is about 23000 lines of FORTRAN. After having discussed some of the details, I shall outline the structure of the GROW module. The first step in implementing the formulae derived in chapter 3 is to choose a basis. This is the subject of the first section.

6.1 Choosing a basis

In chapter 3 we talked about expanding the Hamiltonian and Green function in basis functions. These basis functions we will have to choose before we can implement the method into a computer code, because the calculation of the Hamiltonian matrix elements depends on this basis.

In the Surface Embedded Green Function (SEGF) code, which already implemented a simpler form of the embedding formalism, the basis functions were linear augmented plane waves (LAPW's) as they were originally defined[11]. These LAPW's however seemed to be prone to 'ghost states'. These ghost states arise as spurious poles of the Green function and are not necessarily the ghost states described by other people. Let us examine these ghost states and check how we can adapt the LAPW's to get rid of (most of) them.

6.1.1 Interlude on ghost states

In this section I shall give an explanation for the appearance of spurious states known as *ghost* states. I shall also show that in our calculations based on numerical evaluation of approximated expressions we can never fully get rid of them.

To illustrate this, we take an arbitrary system. This system has, of course, an exact Hamiltonian, $\mathbf{H}_e$. We may not be able to write it down explicitly, but this exact Hamiltonian exists for each system. For this system we can expand the wavefunctions in terms of an infinitely large, complete, basis set $\{\phi_i\}$. This exact basis has an overlap between basis functions, $\mathbf{O}_e$. If we now write the wavefunction of this system as an expansion in the infinite basis:

$$\psi_e = \sum_{i=1}^{\infty} a_i \phi_i \tag{6.1}$$

where the $\{a_i\}$ are the coefficients of the basis functions, we can write the Schrödinger equation as:

$$\mathbf{H}_e\psi_e = E_e\mathbf{O}_e\psi_e \tag{6.2}$$

This is still exact, as no approximations have been made so far. If we define the Green function as:

$$G_e(\mathbf{r},\mathbf{r}') = \sum_{\substack{i=1\\j=1}}^{\infty} g_{ij}\phi_i(\mathbf{r})\phi_j^*(\mathbf{r}') \tag{6.3}$$

we can show that the matrix elements of the $\infty \times \infty$ Green function matrix are given by:

$$g_{ij} = [\mathbf{H}_e - E_e\mathbf{O}_e]^{-1}_{ij} \tag{6.4}$$

From this we see that this Green function has singularities (poles) for those energies that are eigenvalues of the Schrödinger equation (6.2).

But since our resources and our knowledge are limited, we never solve the Schrödinger equation exactly, except for *very* simple systems. Because of our limited knowledge we make approximations to the Hamiltonian, so that we use:

$$\mathbf{H} = \mathbf{H}_e + \Delta\mathbf{H} \tag{6.5}$$

where $\Delta\mathbf{H}$ is the difference between the exact Hamiltonian and our approximated Hamiltonian. Because of our limited resources we sometimes further approximate the Hamiltonian by using a finite basis set. We also inevitably use an inexact representation of the overlap:

$$\mathbf{O} = \mathbf{O}_e + \Delta\mathbf{O} \tag{6.6}$$

which leads to the approximate Schrödinger matrix equation which we solve in our programs:

$$\mathbf{H}\psi = E\mathbf{O}\psi \tag{6.7}$$

Eventually this leads to matrix elements of the finite Green function, given by:

$$g_{ij} = [(\mathbf{H}_e + \Delta\mathbf{H}) - E\,(\mathbf{O}_e + \Delta\mathbf{O})]^{-1}_{ij} \tag{6.8}$$

We can now write down the following expression for the Green function operator:

$$\begin{aligned} G &= [\mathbf{H}_e - E\mathbf{O}_e + \Delta(E)]^{-1} \qquad \Delta(E) = \Delta\mathbf{H} - E\Delta\mathbf{O} \\ &= \left[G_e^{-1} + \Delta(E)\right]^{-1} \\ &= \left[G_e\,(1 + \Delta(E)G_e)^{-1}\right] \\ &= \left[G_e - G_e\Delta(E)G_e(1 + \Delta(E)G_e)^{-1}\right] \end{aligned} \tag{6.9}$$

where the term $\Delta G = G_e\Delta(E)G_e(1 + \Delta(E)G_e)^{-1}$ can give some extra poles, whenever $1 + \Delta(E)G_e = 0$, at energies where there are no physical poles. This is not a proof, and the fact that the exact Hamiltonian is usually energy dependent, and that we linearize the basis or make the Hamiltonian energy independent has some effects as well.

So the approximations we make to the Hamiltonian, plus the use of the finite basis and limited numerical accuracy of our computers may not only lead to a slight shift in the eigenvalues of the Schrödinger equation, but can sometimes lead to poles in the Green function at energies where there should not be any. So in general, we would like our basis functions, and the quantities calculated using them to be as exact as possible.

Now what does this have to do with the 'old' LAPW's? To understand how this argument affects our choice of basis, we will have to look at the definition of LAPW's as used in the SEGF program and most other LAPW methods[11].

6.1.2 The original definition of the LAPW basis

The idea of augmentation was first introduced in the augmented plane wave basis. Since the wavefunction varies rapidly inside the atomic core, an enormous number of plane waves is needed to describe this behaviour in terms of plane waves only. Thus the idea of augmentation is to use different basis functions inside the atom, matching these augmentation functions to the plane wave basis function outside a certain radius.
The original LAPW's were defined as:

$$\phi_{\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r}) = \begin{cases} e^{\imath(\mathbf{g}+\mathbf{k}_\parallel)\,\mathbf{r}} & \mathbf{r} \text{ in the interstitial} \\ \sum_{L}^{n_l} \left[A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} u_l(r_\alpha) + B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} \dot{u}_l(r_\alpha)\right] P_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(\widehat{\mathbf{r}_\alpha}) & \mathbf{r} \text{ in muffin-tin } \alpha \end{cases} \tag{6.10}$$

where $L = (l, m)$ labels the angular momenta, and $P_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(\widehat{\mathbf{r}_\alpha})$ is given by:

$$P_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(\widehat{\mathbf{r}_\alpha}) = 4\pi e^{\imath(\mathbf{g}+\mathbf{k}_\parallel)\,\rho_\alpha} \imath^l Y_{lm}^*(\widehat{\mathbf{g}+\mathbf{k}_\parallel}) Y_{lm}(\widehat{\mathbf{r}_\alpha}) \tag{6.11}$$

The vector ρ_α is the position vector of the centre of muffin-tin α. The index α on $\mathbf{r}$ indicates that this is the position vector relative to ρ_α. The summation over angular quantum numbers up to n_l is typically taken to be 8 or 9. The functions $u_l(r)$ are the solutions of the atomic Schrödinger equation at a certain energy E_l, and $\dot{u}_l(r)$ are the energy derivates of those solutions at E_l (see appendix B). This leads to an energy-independent LAPW, with an energy-linearized solution of the atomic Schrödinger equation. The parameters $A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}$ and $B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}$ are used to match the augmentation function to the plane wave outside the muffin-tin and are therefore different for each l, α, $\mathbf{g}$ and $\mathbf{k}_\parallel$. This matching is done by expanding the plane wave in the interstitial into spherical waves around the atomic centre:

$$\begin{aligned} e^{\imath(\mathbf{g}+\mathbf{k}_\parallel)\,\mathbf{r}} &= e^{\imath(\mathbf{g}+\mathbf{k}_\parallel)\,(\rho_\alpha+\mathbf{r}_\alpha)} \\ &= e^{\imath(\mathbf{g}+\mathbf{k}_\parallel)\,\rho_\alpha} \sum_{L} 4\pi \imath^l j_l(|\mathbf{g}+\mathbf{k}_\parallel| r_\alpha) Y_{l,m}^*(\widehat{\mathbf{g}+\mathbf{k}_\parallel}) Y_{l,m}(\widehat{\mathbf{r}_\alpha}) \\ &= \sum_{L} P_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(\widehat{\mathbf{r}_\alpha}) j_l(|\mathbf{g}+\mathbf{k}_\parallel| r_\alpha) \end{aligned} \tag{6.12}$$

On the surface of the muffin-tin we match the basis function inside and outside both in

amplitude and derivative, and the resulting expressions for $A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}$ and $B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}$ are:

$$A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} = \left. \frac{\frac{\partial u_l(r)}{\partial r} j_l(|\mathbf{g}+\mathbf{k}_\parallel|r) - \dot{u}_l(r) \frac{\partial j_l(|\mathbf{g}+\mathbf{k}_\parallel|r)}{\partial r}}{u_l(r)\frac{\partial u_l(r)}{\partial r} - \dot{u}_l(r)\frac{\partial u_l(r)}{\partial r}} \right|_{r=r_{MT}^\alpha} \tag{6.13a}$$

$$B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} = \left. \frac{u_l(r)\frac{\partial j_l(|\mathbf{g}+\mathbf{k}_\parallel|r)}{\partial r} - \frac{\partial u_l(r)}{\partial r} j_l(|\mathbf{g}+\mathbf{k}_\parallel|r)}{u_l(r)\frac{\partial \dot{u}_l(r)}{\partial r} - \dot{u}_l(r)\frac{\partial u_l(r)}{\partial r}} \right|_{r=r_{MT}^\alpha} \tag{6.13b}$$

where r_{MT}^α is the radius of muffin-tin α.

So, "What is wrong with these LAPW's?". The answer to that question is: "Nothing ..... but .....". As a basis function there is nothing wrong with the LAPW as defined in 6.10, otherwise they would not continue to be used successfully[12]. However, when we want to avoid ghost states as much as possible, there are some minor details of the "old" LAPW which can be improved. The matching on the muffin-tin sphere is done exactly (see (6.13)) only up to a certain angular quantum number n_l. What does this mean? This means that for l greater than this chosen number, the basis functions do not match. Could it be that this mismatch is responsible for some of the ghost states? The answer is "yes". In an earlier study [4] I have shown that inaccuracies like this in the basis functions lead to ghost states.

In the next section I shall make adaptions in the definition of the LAPW which will cure this mismatch problem, the adapted LAPW I call the "new" LAPW.

6.1.3 The new LAPW basis

The obvious solution to the mismatch problem in the linear augmented plane wave basis function is to take the sum over augmentation functions to infinity, but that cannot be done. Or ..... can it? I use the following definition for the new LAPW basis:

$$\phi_{\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r}) = \begin{cases} e^{\imath(\mathbf{g}+\mathbf{k}_\parallel)\cdot\mathbf{r}} & \mathbf{r} \text{ in the interstitial} \\ \sum_L^\infty k_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(r_\alpha) P_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(\widehat{\mathbf{r}_\alpha}) & \mathbf{r} \text{ in muffin-tin } \alpha \end{cases} \tag{6.14}$$

with the following definition for $k_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(r_\alpha)$:

$$k_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(r_\alpha) = \begin{cases} A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} u_l(r_\alpha) + B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} \dot{u}_l(r_\alpha) & \text{for } l \le n_l \\ j_l(|(\mathbf{g}+\mathbf{k}_\parallel)|r_\alpha) & \text{for } l > n_l \end{cases} \tag{6.15}$$

Of course, we cannot really perform this summation to infinity, but we can write (6.14) in such a way that we avoid having to perform this summation explicitly. Using the expansion of the plane wave in terms of spherical waves as given in (6.12), we rewrite (6.14) as:

$$\phi_{\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r}) = \begin{cases} e^{\imath(\mathbf{g}+\mathbf{k}_\parallel)\cdot\mathbf{r}} & \text{for } \mathbf{r} \text{ in the interstitial} \\ e^{\imath(\mathbf{g}+\mathbf{k}_\parallel)\cdot\mathbf{r}} + \sum_L^{n_l} f_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(r_\alpha) P_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(\widehat{\mathbf{r}_\alpha}) & \text{for } \mathbf{r} \text{ in muffin-tin } \alpha \end{cases} \tag{6.16}$$

with:

$$f_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(r_\alpha) = A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} u_l(r_\alpha) + B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} \dot{u}_l(r_\alpha) - j_l(|(\mathbf{g}+\mathbf{k}_\parallel)|r_\alpha) \tag{6.17}$$

The new matching condition is that $f_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(r)$ must be zero both in amplitude and derivative at the muffin-tin radius. Closer examination of the exact form of $f_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(r_\alpha)$ shows us that this leads to the same conditions for $A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}$ and $B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}$ as given in (6.13).

The resulting LAPW is completely matched in amplitude and derivate for *all* l. In hindsight this seems to be a very logical choice for the augmentation, because only those parts of the plane wave that have to incorporate the behaviour near the atomic core are augmented. This also means that in this new LAPW we can choose n_l such that we only include physical l values, i.e. typically $l = 0..2$ or 3.

Tests with this new definition of the LAPW in the old SEGF code have shown that this LAPW cures almost all ghost state problems. However, the reason why most other people may have decided not to use this definition of the LAPW will become clear in the next section; this new LAPW doesn't make the Hamiltonian matrix elements any simpler, on the contrary. These Hamiltonian matrix elements will be derived in section 6.2, but first I shall briefly discuss the LAPW's as they are implemented in the computer code.

6.1.4 The LAPW basis in the computer code

In the computer code based on the embedding method for layered systems as described in chapter 3, we use an LAPW basis which is slightly different from the one described above. Because the z direction is treated on a different footing from the x and y directions, we use the following definition for the LAPW:

$$\phi_{\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r}) = e^{i(\mathbf{G}+\mathbf{k}_\parallel)\cdot\mathbf{R}} \left\{ \begin{array}{c} \sin(g_z z) \\ \cos(g_z z) \end{array} \right\} + \sum_L^{n_l} f_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(r_\alpha) M_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(\widehat{\mathbf{r}_\alpha}) \theta(\mathbf{r}_\alpha - \rho_\alpha) \tag{6.18}$$

where $\theta(\mathbf{r}_\alpha - \rho_\alpha)$ equals unity when inside muffin-tin α, and zero everywhere else. As is clear from this we always use the fact that the out-of-plane basis vector is along the z-axis. The sine is used when g_z is an odd multiple of the reciprocal basis vector g_3, the z-component of the reciprocal basis, and the cosine when g_z is an even multiple of g_3. The augmentation function $f_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(r_\alpha)$ is the same as defined in (6.17). However, since this originally matched on to an exponential function in the interstitial region, the $\mathrm{P}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}$ has to be adapted to take care of the fact that we now have a sine or cosine in the z-direction. This leads to:

$$\mathrm{M}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} = 4\pi i^l f_t \left(f_1 e^{i\mathbf{g}^+\cdot\mathbf{r}_\alpha} Y_L(\widehat{\mathbf{g}^+}) + f_2 e^{i\mathbf{g}^-\,\mathbf{r}_\alpha} Y_L(\widehat{\mathbf{g}^-}) \right) \tag{6.19}$$

$$= 4\pi i^l e^{i\mathbf{g}\,\mathbf{R}_\alpha} f_t \left(f_1 e^{ig_z z_\alpha} Y_L(\widehat{\mathbf{g}^+}) + f_2 e^{-ig_z z_\alpha} Y_L(\widehat{\mathbf{g}^-}) \right) \tag{6.20}$$

where $\mathbf{g}^+ = (\mathbf{G}, g_z)$, $\mathbf{g}^- = (\mathbf{G}, -g_z)$, and f_t, f_1 and f_2 take the following values:

z-var	f_t	f_1	f_2
sin	$\frac{1}{2i}$	1	-1
cos	$\frac{1}{2}$	1	1

(6.20) has been derived by writing the sine and cosine as linear combinations of exponentials. In the remainder of this chapter we will derive expressions using the LAPW definition in (6.16)

because that is less complicated. The final expressions were adapted before implementation in the code, by replacing $P_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}$ with $M_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}$ and taking care to replace exponentials with the planar exponential combined with a sine or cosine.

6.2 Hamiltonian matrix elements

To simplify derivation and eventual coding, we break up the expressions for the Hamiltonian matrix elements into manageable pieces. We start in the muffin-tin, with a simple spherical potential. But since we want to perform full-potential linear augmented plane wave (FLAPW) calculations, we then consider non-spherical contributions inside the muffin-tin, and then deviations from constant potential in the interstitial region.

6.2.1 Muffin-tin terms with a spherical potential

The energy-dependent semi-relativistic Hamiltonian inside muffin-tin α can be written as[13, 14]:

$$H(E) = -\frac{1}{2M}\frac{1}{r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left(r^2\frac{\mathrm{d}}{\mathrm{d}r}\right) + \frac{l(l+1)}{2Mr^2} + V(r) - \frac{V'(r)}{4M^2c^2}\frac{\mathrm{d}}{\mathrm{d}r} \tag{6.21}$$

where $V'(r)$ is the radial derivative of the potential, and $M(r)$ is the relativistic mass, given by:

$$M(r) = m + \frac{1}{2c^2}\left(E - V(r)\right) \tag{6.22}$$

The non-relativistic free-electron Hamiltonian is given by:

$$H_0(r) = -\frac{1}{2m}\frac{1}{r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left(r^2\frac{\mathrm{d}}{\mathrm{d}r}\right) + \frac{l(l+1)}{2mr^2} \tag{6.23}$$

Up to order $\frac{1}{c^2}$, H(E) can be written in terms of H_0, as:

$$H(E) = H_0 + V(r) - \frac{(E - V(r))}{2mc^2}H_0 - \frac{V'(r)}{4m^2c^2}\frac{\mathrm{d}}{\mathrm{d}r} \tag{6.24}$$

We also know that:

$$H_0 e^{i\mathbf{g}\cdot\mathbf{r}} = \tfrac{1}{2}|\mathbf{g}|^2 e^{i\mathbf{g}\cdot\mathbf{r}} \tag{6.25a}$$

$$H_0 j_l(|\mathbf{g}|r) = \tfrac{1}{2}|\mathbf{g}|^2 j_l(|\mathbf{g}|r) \tag{6.25b}$$

From here on, we will use $\tilde{\mathbf{g}}$ to represent $\mathbf{g} + \mathbf{k}_\parallel$ and $\tilde{\mathbf{g}}'$ to represent $\mathbf{g}' + \mathbf{k}_\parallel$. When working

out the matrix elements of H in terms of our LAPW-basis functions, we obtain:

$$\begin{aligned}\left\langle \phi_{\mathbf{g}}^{\mathbf{k}_\parallel} \middle| H(E) \middle| \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel} \right\rangle = \left\langle e^{i\tilde{\mathbf{g}}\,\mathbf{r}_\alpha} \middle| H(E) \middle| e^{i\tilde{\mathbf{g}}'\,\mathbf{r}_\alpha} \right\rangle + \sum_{L\alpha}^{n_l} Q_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{L,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \\ \Big[A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} A_{l,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \cdot \int_\alpha r^2 \mathrm{d}r u_{l,\alpha}(r) H(E) u_{l,\alpha}(r) \\ + A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} B_{l,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \cdot \int_\alpha r^2 \mathrm{d}r u_{l,\alpha}(r) H(E) \dot{u}_{l,\alpha}(r) \\ + B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} A_{l,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \cdot \int_\alpha r^2 \mathrm{d}r \dot{u}_{l,\alpha}(r) H(E) u_{l,\alpha}(r) \\ + B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} B_{l,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \cdot \int_\alpha r^2 \mathrm{d}r \dot{u}_{l,\alpha}(r) H(E) \dot{u}_{l,\alpha}(r) \\ - \int_\alpha r^2 \mathrm{d}r j_l(|\tilde{\mathbf{g}}|r) H(E) j_l(|\tilde{\mathbf{g}}'|r) \Big] \end{aligned} \tag{6.26}$$

where all terms involving j_l's and $u_{l,\alpha}$'s or $u_{l,\alpha}$'s have cancelled out. The $Q_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}$'s are the same as the $P_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}$'s with the $Y_L(\widehat{\mathbf{r}_\alpha})$ part integrated out.
We shall now derive expressions for the different terms in equation (6.26). We write $H(E)$ as in equation (6.24) which leads to:

$$\begin{aligned}\left\langle e^{i\tilde{\mathbf{g}}\,\mathbf{r}} \middle| H(E) \middle| e^{i\tilde{\mathbf{g}}'\,\mathbf{r}} \right\rangle =& \left\langle e^{i\tilde{\mathbf{g}}\,\mathbf{r}} \middle| H_0 \middle| e^{i\tilde{\mathbf{g}}'\,\mathbf{r}} \right\rangle + - \left\langle e^{i\tilde{\mathbf{g}}\,\mathbf{r}} \middle| \frac{(E - V(r))}{2mc^2} H_0 \middle| e^{i\tilde{\mathbf{g}}'\,\mathbf{r}} \right\rangle \\ &+ \left\langle e^{i\tilde{\mathbf{g}}\,\mathbf{r}} \middle| V(r) \middle| e^{i\tilde{\mathbf{g}}'\,\mathbf{r}} \right\rangle - \left\langle e^{i\tilde{\mathbf{g}}\,\mathbf{r}} \middle| \frac{V'(r)}{4m^2c^2} \frac{\mathrm{d}}{\mathrm{d}r} \middle| e^{i\tilde{\mathbf{g}}'\,\mathbf{r}} \right\rangle \end{aligned} \tag{6.27}$$

These terms can be written as:

$$\begin{aligned}\left\langle e^{i\tilde{\mathbf{g}}\,\mathbf{r}} \middle| H_0 \middle| e^{i\tilde{\mathbf{g}}'\,\mathbf{r}} \right\rangle =& \left\langle e^{i\tilde{\mathbf{g}}\,\mathbf{r}} \middle| \tfrac{1}{2}|\tilde{\mathbf{g}}|^2 \middle| e^{i\tilde{\mathbf{g}}'\,\mathbf{r}} \right\rangle \\ =& \tfrac{1}{2}|\tilde{\mathbf{g}}|^2 \mathrm{O}_{\mathbf{g}\mathbf{g}'}^{\mathrm{PW}} \end{aligned} \tag{6.28}$$

$$\left\langle e^{i\tilde{\mathbf{g}}\,\mathbf{r}} \middle| \frac{(E - V(r))}{2mc^2} H_0 \middle| e^{i\tilde{\mathbf{g}}'\,\mathbf{r}} \right\rangle = \frac{E}{2mc^2} \left\langle e^{i\tilde{\mathbf{g}}\,\mathbf{r}} \middle| H_0 \middle| e^{i\tilde{\mathbf{g}}'\,\mathbf{r}} \right\rangle - \frac{1}{2mc^2} \left\langle e^{i\tilde{\mathbf{g}}\,\mathbf{r}} \middle| V(r) H_0 \middle| e^{i\tilde{\mathbf{g}}'\,\mathbf{r}} \right\rangle$$

$$= \frac{E|\tilde{\mathbf{g}}'|^2}{4mc^2} \mathrm{O}_{\mathbf{g}\mathbf{g}'}^{\mathrm{PW}} - \frac{|\tilde{\mathbf{g}}'|^2}{4mc^2} \mathrm{V}_{\mathbf{g}\mathbf{g}'}^{\mathrm{PW}} \tag{6.29}$$

$$\left\langle e^{i\tilde{\mathbf{g}}\,\mathbf{r}} \middle| V(r) \middle| e^{i\tilde{\mathbf{g}}'\,\mathbf{r}} \right\rangle = \mathrm{V}_{\mathbf{g}\mathbf{g}'}^{\mathrm{PW}} \tag{6.30}$$

$$\left\langle e^{i\tilde{\mathbf{g}}\,\mathbf{r}} \middle| \frac{V'(r)}{4m^2c^2} \frac{\mathrm{d}}{\mathrm{d}r} \middle| e^{i\tilde{\mathbf{g}}'\,\mathbf{r}} \right\rangle = \frac{1}{4m^2c^2} \int \mathrm{d}^3\mathbf{r}\, e^{-i\tilde{\mathbf{g}}\,\mathbf{r}} \left[V'(r) \frac{\mathrm{d}}{\mathrm{d}r} \right] e^{i\tilde{\mathbf{g}}\,\mathbf{r}} \tag{6.31}$$

The term defined in equation (6.31) is usually called the Darwin term. When we expand the plane waves around the origin of the MT into Bessel functions, and integrate by parts, we

end up with the following expression for (6 31):

$$\left\langle e^{i\tilde{\mathbf{g}}\,\mathbf{r}}\left|\frac{V'(r)}{4m^2c^2}\frac{\mathrm{d}}{\mathrm{d}r}\right|e^{i\tilde{\mathbf{g}}'\,\mathbf{r}}\right\rangle = -\frac{|\tilde{\mathbf{g}}'|^2}{4m^2c^2}\sum_{\alpha}\sum_{L}^{\infty} Q_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{L,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times$$
$$\left[2\int_\alpha r\mathrm{d}r\; V(r)j_l(|\tilde{\mathbf{g}}|r)j_l'(|\tilde{\mathbf{g}}'|r)\right.$$
$$+|\tilde{\mathbf{g}}|\int_\alpha r^2\mathrm{d}r\; V(r)j_l'(|\tilde{\mathbf{g}}|r)j_l'(|\tilde{\mathbf{g}}'|r)$$
$$\left.+|\tilde{\mathbf{g}}'|\int_\alpha r^2\mathrm{d}r\; V(r)j_l(|\tilde{\mathbf{g}}|r)j_l''(|\tilde{\mathbf{g}}'|r)\right] \quad (6.32)$$

In a practical evaluation of this Darwin term, the infinite sum over L in (6.32) can only be done up to some maximum L.
The expression $O_{\mathbf{g},\mathbf{g}'}^{\mathrm{PW}}$ as introduced in (6.28) can be evaluated as follows:

$$O_{\mathbf{g},\mathbf{g}'}^{\mathrm{PW}} = 4\pi\sum_\alpha\int_\alpha r^2\mathrm{d}r\; j_0(|\mathbf{g}-\mathbf{g}'|r) \quad (6.33)$$

A similar expression holds for $V_{\mathbf{g},\mathbf{g}'}^{\mathrm{PW}}$, introduced in (6.29)

$$V_{\mathbf{g},\mathbf{g}'}^{\mathrm{PW}} = 4\pi\sum_\alpha\int_\alpha r^2\mathrm{d}r\; j_0(|\mathbf{g}-\mathbf{g}'|r)V(r) \quad (6.34)$$

Before we can do something with the remaining terms in (6 26), we want to have expressions for the integrals containing $u_l(r)$ and $\dot{u}_l(r)$. These integrals are derived in appendix B.
To calculate the integral in the last term of (6.26) we write $H(E)$ as in (6.24), so that we get:

$$\int_\alpha r^2\mathrm{d}r\; j_l(|\tilde{\mathbf{g}}|r)H(E)j_l(|\tilde{\mathbf{g}}'|r) =$$
$$\int_\alpha r^2\mathrm{d}r\; j_l(|\tilde{\mathbf{g}}|r)H_0 j_l(|\tilde{\mathbf{g}}'|r)$$
$$-\int_\alpha r^2\mathrm{d}r\; j_l(|\tilde{\mathbf{g}}|r)\frac{E-V(r)}{2mc^2}H_0 j_l(|\tilde{\mathbf{g}}'|r)$$
$$+\int_\alpha r^2\mathrm{d}r\; j_l(|\tilde{\mathbf{g}}|r)V(r)j_l(|\tilde{\mathbf{g}}'|r)$$
$$-\int_\alpha r^2\mathrm{d}r\; j_l(|\tilde{\mathbf{g}}|r)\frac{V'(r)}{4m^2c^2}\frac{\mathrm{d}}{\mathrm{d}\,r}j_l(|\tilde{\mathbf{g}}'|r) \quad (6.35)$$

The last term in (6.35) also shows up in the derivation of (6.32); in fact it can be shown that this term will cancel with the first n_l terms in the infinite sum in (6.32).
Substituting (6.25b) into the first term of the right-hand side of (6.35) we obtain:

$$\int_\alpha r^2\mathrm{d}r\; j_l(|\tilde{\mathbf{g}}|r)H_0 j_l(|\tilde{\mathbf{g}}'|r) = \tfrac{1}{2}|\tilde{\mathbf{g}}'|^2\int_\alpha r^2\mathrm{d}r\; j_l(|\tilde{\mathbf{g}}|r)j_l(|\tilde{\mathbf{g}}'|r) \quad (6.36)$$

Similarly for the second term of the right-hand side of (6.35):

$$-\int_\alpha r^2\mathrm{d}r\; j_l(|\tilde{\mathbf{g}}|r)\frac{E-V(r)}{2mc^2}H_0 j_l(|\tilde{\mathbf{g}}'|r) =$$
$$-\frac{E|\tilde{\mathbf{g}}'|^2}{4mc^2}\int_\alpha r^2\mathrm{d}r\; j_l(|\tilde{\mathbf{g}}|r)j_l(|\tilde{\mathbf{g}}'|r)$$
$$+\frac{|\tilde{\mathbf{g}}'|^2}{4mc^2}\int_\alpha r^2\mathrm{d}r\; j_l(|\tilde{\mathbf{g}}|r)V(r)j_l(|\tilde{\mathbf{g}}'|r) \quad (6.37)$$

Now we define:

$$\mathrm{O}^{l,\alpha}_{\tilde{\mathbf{g}},\tilde{\mathbf{g}}'} = \int_\alpha r^2 \mathrm{d}r \; j_l(|\tilde{\mathbf{g}}|r) j_l(|\tilde{\mathbf{g}}'|r) \tag{6.38a}$$

$$\mathrm{V}^{l,\alpha}_{\tilde{\mathbf{g}},\tilde{\mathbf{g}}'} = \int_\alpha r^2 \mathrm{d}r \; j_l(|(\tilde{\mathbf{g}}|r) V(r) j_l(|\tilde{\mathbf{g}}'|r) \tag{6.38b}$$

and we can write (6.35) as:

$$\int_\alpha r^2 \mathrm{d}r \; j_l(|\tilde{\mathbf{g}}|r) H j_l(|\tilde{\mathbf{g}}'|r) = \left\{ \tfrac{1}{2}|\tilde{\mathbf{g}}'|^2 - \frac{E|\tilde{\mathbf{g}}'|^2}{4mc^2} \right\} \mathrm{O}^{l,\alpha}_{\tilde{\mathbf{g}},\tilde{\mathbf{g}}'} + \left\{ 1 + \frac{|\tilde{\mathbf{g}}'|^2}{4mc^2} \right\} \mathrm{V}^{l,\alpha}_{\tilde{\mathbf{g}},\tilde{\mathbf{g}}'} + \text{Darwin terms} \tag{6.39}$$

Although the LAPW basis functions are energy-independent, the energy-dependence of the semi-relativistic term leads to a Hamiltonian which is linear in energy (up to order $\frac{1}{c^2}$):

$$H(E) = H_1 + H_2 \cdot E + O(\frac{1}{c^4}) \tag{6.40}$$

We collect the terms of (6.26) which have been worked out on the previous pages and in appendix B, and group all those terms independent of E into H_1 and all terms dependent on E into H_2 to obtain the following expressions for H_1 and H_2:

$$\begin{aligned} H_1 = {} & \tfrac{1}{2}|\tilde{\mathbf{g}}'|^2 \mathrm{O}^{\mathrm{PW}}_{\mathbf{g},\mathbf{g}'} + \left(1 + \frac{|\tilde{\mathbf{g}}'|^2}{4mc^2} \right) \mathrm{V}^{\mathrm{PW}}_{\mathbf{g},\mathbf{g}'} \\ & + \sum_\alpha \sum_L^{n_l} \mathrm{Q}^{\mathbf{g},\mathbf{k}_\parallel *}_{L,\alpha} \mathrm{Q}^{\mathbf{g}',\mathbf{k}_\parallel}_{L,\alpha} \times \Big[A^{\mathbf{g},\mathbf{k}_\parallel}_{l,\alpha} A^{\mathbf{g}',\mathbf{k}_\parallel}_{l,\alpha} \times \left\{ E_l + \frac{E_l}{2mc^2} \langle u_{l,\alpha}|H_0|u_{l,\alpha}\rangle \right\} \\ & \qquad A^{\mathbf{g},\mathbf{k}_\parallel}_{l,\alpha} B^{\mathbf{g}',\mathbf{k}_\parallel}_{l,\alpha} \times \left\{ 1 + \frac{E_l}{2mc^2} - \right. \\ & \qquad \left. \frac{1}{2mc^2} \langle u_{l,\alpha}|V|u_{l,\alpha}\rangle + \frac{E_l}{2mc^2} \langle u_{l,\alpha}|H_0|\dot{u}_{l,\alpha}\rangle \right\} \\ & \qquad B^{\mathbf{g},\mathbf{k}_\parallel}_{l,\alpha} A^{\mathbf{g}',\mathbf{k}_\parallel}_{l,\alpha} \times \left\{ \frac{E_l}{2mc^2} \langle \dot{u}_{l,\alpha}|H_0|u_{l,\alpha}\rangle \right\} \\ & \qquad B^{\mathbf{g},\mathbf{k}_\parallel}_{l,\alpha} B^{\mathbf{g}',\mathbf{k}_\parallel}_{l,\alpha} \times \left\{ -\frac{1}{2mc^2} \langle \dot{u}_{l,\alpha}|V|u_{l,\alpha}\rangle + \right. \\ & \qquad \left. E_l \langle \dot{u}_{l,\alpha}|\dot{u}_{l,\alpha}\rangle + \frac{E_l}{2mc^2} \langle \dot{u}_{l,\alpha}|H_0|\dot{u}_{l,\alpha}\rangle \right\} \\ & \qquad - \left\{ \tfrac{1}{2}|\tilde{\mathbf{g}}'|^2 \right\} \mathrm{O}^{l,\alpha}_{\tilde{\mathbf{g}},\tilde{\mathbf{g}}'} - \left\{ 1 + \frac{|\tilde{\mathbf{g}}'|^2}{4mc^2} \right\} \mathrm{V}^{l,\alpha}_{\tilde{\mathbf{g}},\tilde{\mathbf{g}}'} \Big] \end{aligned}$$

$$+\sum_{\alpha}\sum_{L=n_l+1}^{\infty} Q_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{L,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \left[\frac{|\tilde{\mathbf{g}}'|}{4m^2c^2} \times \left\{ r_{MT}^{\alpha\,2} j_l(|\tilde{\mathbf{g}}| r_{MT}^{\alpha}) j_l'(|\tilde{\mathbf{g}}'| r_{MT}^{\alpha}) V(r_{MT}^{\alpha}) \right\} - \frac{|\tilde{\mathbf{g}}'|}{4m^2c^2} \times \left\{ 2\int_{\alpha} r\mathrm{d}r\, V(r) j_l(|\tilde{\mathbf{g}}| r) j_l'(|\tilde{\mathbf{g}}'| r) \right\} + \frac{|\tilde{\mathbf{g}}'||\tilde{\mathbf{g}}|}{4m^2c^2} \times \left\{ \int_{\alpha} r^2\mathrm{d}r\, V(r) j_l'(|\tilde{\mathbf{g}}| r) j_l'(|\tilde{\mathbf{g}}'| r) \right\} + \frac{|\tilde{\mathbf{g}}'|^2}{4m^2c^2} \times \left\{ \int_{\alpha} r^2\mathrm{d}r\, V(r) j_l(|\tilde{\mathbf{g}}| r) j_l''(|\tilde{\mathbf{g}}'| r) \right\} \right] \tag{6 41}$$

$$H_2 = -\frac{|\mathbf{g}'|^2}{4mc^2} O_{\mathbf{g},\mathbf{g}'}^{\mathrm{PW}} + \sum_{\alpha}\sum_{L}^{n_l} Q_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{L,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \left[A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} A_{l,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \left\{ -\frac{1}{2mc^2} \langle u_{l,\alpha} | H_0 | u_{l,\alpha} \rangle \right\} + A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} B_{l,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \left\{ -\frac{1}{2mc^2} \langle u_{l,\alpha} | H_0 | u_{l,\alpha} \rangle \right\} + B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} A_{l,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \left\{ -\frac{1}{2mc^2} \langle u_{l,\alpha} | H_0 | u_{l,\alpha} \rangle \right\} + B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} B_{l,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \left\{ -\frac{1}{2mc^2} \langle u_{l,\alpha} | H_0 | u_{l,\alpha} \rangle \right\} + \frac{|\tilde{\mathbf{g}}'|^2}{4mc^2} O_{\tilde{\mathbf{g}},\tilde{\mathbf{g}}'}^{l,\alpha} \right] \tag{6 42}$$

In (6 41) and (6 42) we have derived expressions for H_1 and H_2, which are both independent of energy These matrix elements can be implemented in a computer code, after being adapted to our sine/cosine LAPW as defined in (6 18)

6.2.2 Muffin-tin terms with a non-spherical potential

In general the potential in the muffin-tin is not spherically symmetric One important example is for an atom at a surface, where a spherically symmetric potential would be a rather bad description of reality We use the general expansion of the potential in the muffin-tin as given by

$$V(\mathbf{r}_\alpha) = \sum_{L}^{n_l} V_L(r_\alpha) Y_L(\hat{\mathbf{r}_\alpha}) \tag{6 43}$$

where $L = (l, m)$ The potential $V(r_\alpha)$ we have treated in the previous section is of course the $l = 0, m = 0$ part of the potential The non-spherical part of the potential has to be added to (6 21), we leave the semi-relativistic corrections as they are, because these are only important near the nucleus where spherically symmetric terms dominate

This means that we have to find the matrix elements in the LAPW basis for the following correction to the Hamiltonian

$$\Delta H = \sum_{L=1}^{\tilde{n}_l} V_L(r) Y_L(\hat{r}) \tag{6 44}$$

The derivation is similar to (6.26), but not all terms drop out. The matrix elements are of the form:

$$\left\langle \phi_{\mathbf{g}}^{\mathbf{k}_\parallel} \middle| \Delta H \middle| \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel} \right\rangle = \left\langle e^{i\tilde{\mathbf{g}}\,\mathbf{r}} \middle| \Delta H \middle| e^{i\tilde{\mathbf{g}}'\,\mathbf{r}} \right\rangle + \sum_{\alpha} \sum_{L}^{n_l} \sum_{L'}^{n_l} \text{terms of type I} +$$
$$\sum_{\alpha} \sum_{L=n_l+1}^{\infty} \sum_{L'}^{n_l} \text{terms of type II} + \sum_{\alpha} \sum_{L}^{n_l} \sum_{L'=n_l+1}^{\infty} \text{terms of type III} \quad (6.45)$$

Terms of type I are terms as in (6.26), where both functions in the integral are of the same type, i.e. both atomic solutions or both Bessel functions. Terms of type II and III are the mixed terms which dropped out in (6.26) because of exact cancellation. We will call the first sum in (6.45) I_1, the second sum I_2, and the third I_3. These are defined as:

$$\begin{aligned} I_1 = \sum_{\alpha} \sum_{L}^{n_l} \sum_{L'}^{n_l} \Big[& A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} A_{l',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \int_{\alpha} \mathrm{d}^3\mathbf{r} u_{l,\alpha}^*(r) \mathrm{P}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *}(\hat{\mathbf{r}}) \Delta H u_{l',\alpha}(r) \mathrm{P}_{L',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel}(\hat{\mathbf{r}}) \\ & + A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} B_{l',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \int_{\alpha} \mathrm{d}^3\mathbf{r} u_{l,\alpha}^*(r) \mathrm{P}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *}(\hat{\mathbf{r}}) \Delta H \dot{u}_{l',\alpha}(r) \mathrm{P}_{L',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel}(\hat{\mathbf{r}}) \\ & + B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} A_{l',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \int_{\alpha} \mathrm{d}^3\mathbf{r} \dot{u}_{l,\alpha}^*(r) \mathrm{P}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *}(\hat{\mathbf{r}}) \Delta H u_{l',\alpha}(r) \mathrm{P}_{L',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel}(\hat{\mathbf{r}}) \\ & + B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} B_{l',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \int_{\alpha} \mathrm{d}^3\mathbf{r} \dot{u}_{l,\alpha}^*(r) \mathrm{P}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *}(\hat{\mathbf{r}}) \Delta H \dot{u}_{l',\alpha}(r) \mathrm{P}_{L',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel}(\hat{\mathbf{r}}) \\ & - \int_{\alpha} \mathrm{d}^3\mathbf{r} j_l^*(|\tilde{\mathbf{g}}|r) \mathrm{P}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *}(\hat{\mathbf{r}}) \Delta H j_{l'}(|\tilde{\mathbf{g}}'|r) \mathrm{P}_{L',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel}(\hat{\mathbf{r}}) \Big] \end{aligned} \quad (6.46)$$

$$\begin{aligned} I_2 = \sum_{\alpha} \sum_{L=n_l+1}^{\infty} \sum_{L'}^{n_l} \Big[& A_{l',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \int_{\alpha} \mathrm{d}^3\mathbf{r} j_l^*(|\tilde{\mathbf{g}}|r) \mathrm{P}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *}(\hat{\mathbf{r}}) \Delta H u_{l',\alpha}(r) \mathrm{P}_{L',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel}(\hat{\mathbf{r}}) \\ & + B_{l',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \int_{\alpha} \mathrm{d}^3\mathbf{r} j_l^*(|\tilde{\mathbf{g}}|r) \mathrm{P}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *}(\hat{\mathbf{r}}) \Delta H \dot{u}_{l',\alpha}(r) \mathrm{P}_{L',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel}(\hat{\mathbf{r}}) \\ & - \int_{\alpha} \mathrm{d}^3\mathbf{r} j_l^*(|\tilde{\mathbf{g}}|r) \mathrm{P}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *}(\hat{\mathbf{r}}) \Delta H j_{l'}(|\tilde{\mathbf{g}}'|r) \mathrm{P}_{L',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel}(\hat{\mathbf{r}}) \Big] \end{aligned} \quad (6.47)$$

$$\begin{aligned} I_3 = \sum_{\alpha} \sum_{L}^{n_l} \sum_{L'=n_l+1}^{\infty} \Big[& A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} \times \int_{\alpha} \mathrm{d}^3\mathbf{r} u_{l,\alpha}^*(r) \mathrm{P}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *}(\hat{\mathbf{r}}) \Delta H j_{l'}(|\tilde{\mathbf{g}}'|r) \mathrm{P}_{L',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel}(\hat{\mathbf{r}}) \\ & + B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} \times \int_{\alpha} \mathrm{d}^3\mathbf{r} \dot{u}_{l,\alpha}^*(r) \mathrm{P}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *}(\hat{\mathbf{r}}) \Delta H j_{l'}(|\tilde{\mathbf{g}}'|r) \mathrm{P}_{L',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel}(\hat{\mathbf{r}}) \\ & - \int_{\alpha} \mathrm{d}^3\mathbf{r} j_l^*(|\tilde{\mathbf{g}}|r) \mathrm{P}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *}(\hat{\mathbf{r}}) \Delta H j_{l'}(|\tilde{\mathbf{g}}'|r) \mathrm{P}_{L',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel}(\hat{\mathbf{r}}) \Big] \end{aligned} \quad (6.48)$$

The integrals over the MT in (6.46), (6.47), and (6.48) contain the following type of angular integrals, where the Y_L's have been taken out of the $\mathrm{P}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}$'s.

$$C_{LL'}^{L''} = \int_{\alpha} \mathrm{d}\hat{\mathbf{r}} Y_L^*(\hat{\mathbf{r}}) Y_{L''}(\hat{\mathbf{r}}) Y_{L'}(\hat{\mathbf{r}}) \quad (6.49)$$

Integrals of this type are usually called Gaunt or Clebsch-Gordon coefficients[13, 14]. First let us examine I_1 as expressed in (6.46). We substitute (6.44) and integrate over angles, to

obtain:

$$I_1 = \sum_{\alpha}\sum_{L}^{n_l}\sum_{L'}^{n_l}\sum_{L''}^{\tilde{n}_l} \mathrm{Q}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel\,*}\mathrm{Q}_{L',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel}\Big[A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel\,*}A_{l',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \int_\alpha r^2\mathrm{d}r u_{l,\alpha}^*(r)V_{L''}(r)u_{l',\alpha}(r)$$
$$+ A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel\,*}B_{l',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \int_\alpha r^2\mathrm{d}r u_{l,\alpha}^*(r)V_{L''}(r)\dot{u}_{l',\alpha}(r)$$
$$+ B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel\,*}A_{l',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \int_\alpha r^2\mathrm{d}r \dot{u}_{l,\alpha}^*(r)V_{L''}(r)u_{l',\alpha}(r)$$
$$+ B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel\,*}B_{l',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \int_\alpha r^2\mathrm{d}r u_{l,\alpha}^*(r)V_{L''}(r)u_{l',\alpha}(r)$$
$$- \int_\alpha r^2\mathrm{d}r j_l^*(|\tilde{\mathbf{g}}|r)V_{L''}(r)j_{l'}(|\tilde{\mathbf{g}}'|r)\Big] \times C_{LL'}^{L''} \qquad (6\,50)$$

Similarly for I_2:

$$I_2 = \sum_{\alpha}\sum_{L=n_l+1}^{\infty}\sum_{L'}^{n_l}\sum_{L''}^{\tilde{n}_l} \mathrm{Q}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel\,*}\mathrm{Q}_{L',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel}\Big[A_{l',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \int_\alpha r^2\mathrm{d}r j_l^*(|\tilde{\mathbf{g}}|r)V_{L''}(r)u_{l',\alpha}(r)$$
$$+ B_{l',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \int_\alpha r^2\mathrm{d}r j_l^*(|\tilde{\mathbf{g}}|r)V_{L''}(r)\dot{u}_{l',\alpha}(r)$$
$$- \int_\alpha r^2\mathrm{d}r j_l^*(|\tilde{\mathbf{g}}|r)V_{L''}(r)j_{l'}(|\tilde{\mathbf{g}}'|r)\Big] \times C_{LL'}^{L''} \qquad (6.51)$$

and I_3:

$$I_3 = \sum_{\alpha}\sum_{L}^{n_l}\sum_{L'=n_l+1}^{\infty}\sum_{L''}^{\tilde{n}_l} \mathrm{Q}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel\,*}\mathrm{Q}_{L',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel}\Big[A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel\,*} \times \int_\alpha r^2\mathrm{d}r u_{l,\alpha}^*(r)V_{L''}(r)j_{l'}(|\tilde{\mathbf{g}}'|r)$$
$$+ B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel\,*} \times \int_\alpha r^2\mathrm{d}r u_{l,\alpha}^*(r)V_{L''}(r)j_{l'}(|\tilde{\mathbf{g}}'|r)$$
$$- \int_\alpha r^2\mathrm{d}r j_l^*(|\tilde{\mathbf{g}}|r)V_{L''}(r)j_{l'}(|\tilde{\mathbf{g}}'|r)\Big] \times C_{LL'}^{L''} \qquad (6\,52)$$

The first term on the right-hand side of (6.45) we write as:

$$\left\langle e^{i\tilde{\mathbf{g}}\,\mathbf{r}}\middle|\Delta H\middle|e^{i\tilde{\mathbf{g}}'\,\mathbf{r}}\right\rangle = \int_\alpha \mathrm{d}^3\mathbf{r}e^{-i\tilde{\mathbf{g}}\,\mathbf{r}}\Delta H e^{i\tilde{\mathbf{g}}'\,\mathbf{r}}$$
$$= 4\pi\sum_{L=1}^{\tilde{n}_l} i^l Y_L^*(\widehat{\mathbf{g}'-\mathbf{g}})\int_\alpha r^2\mathrm{d}r\; j_l(|\mathbf{g}'-\mathbf{g}|r)V_L(r) \qquad (6.53)$$

The $l=0$ term, not included in (6.53), appeared before as the $\mathrm{V}_{\mathbf{g},\mathbf{g}'}^{\mathrm{PW}}$ in (6.34). Combining I_1,

I_2 and I_3 with (6 53), we obtain the total muffin-tin warping potential matrix elements

$$\begin{aligned}
\left\langle \phi_{\mathbf{g}}^{\mathbf{k}_\parallel} \middle| \Delta H \middle| \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel} \right\rangle = 4\pi \sum_{L=1}^{\tilde{n}_l} i^l Y_L^*(\widehat{\mathbf{g}'-\mathbf{g}}) \int_\alpha r^2 \mathrm{d}r \; j_l(|\mathbf{g}'-\mathbf{g}|r) V_L(r) + \\
\sum_\alpha \sum_L^{n_l} \sum_{L'}^{n_l} \sum_{L''=1}^{\tilde{n}_l} \mathrm{Q}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} \mathrm{Q}_{L',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \Big[A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} A_{l',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \int_\alpha r^2 \mathrm{d}r u_{l,\alpha}^*(r) V_{L''}(r) u_{l',\alpha}(r) \\
+ A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} B_{l',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \int_\alpha r^2 \mathrm{d}r u_{l,\alpha}^*(r) V_{L''}(r) u_{l',\alpha}(r) \\
+ B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} A_{l',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \int_\alpha r^2 \mathrm{d}r u_{l,\alpha}^*(r) V_{L''}(r) u_{l',\alpha}(r) \\
+ B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} B_{l',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \int_\alpha r^2 \mathrm{d}r u_{l,\alpha}^*(r) V_{L''}(r) u_{l',\alpha}(r) \\
- \int_\alpha r^2 \mathrm{d}r j_l^*(|\tilde{\mathbf{g}}|r) V_{L''}(r) j_{l'}(|\tilde{\mathbf{g}}'|r) \Big] \times C_{LL'}^{L''} \\
+ \sum_\alpha \sum_{L=n_l+1}^{\infty} \sum_{L'}^{n_l} \sum_{L''=1}^{\tilde{n}_l} \mathrm{Q}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} \mathrm{Q}_{L',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \Big[A_{l',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \int_\alpha r^2 \mathrm{d}r j_l^*(|\tilde{\mathbf{g}}|r) V_{L''}(r) u_{l',\alpha}(r) \\
+ B_{l',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \int_\alpha r^2 \mathrm{d}r j_l^*(|\tilde{\mathbf{g}}|r) V_{L''}(r) u_{l',\alpha}(r) \\
- \int_\alpha r^2 \mathrm{d}r j_l^*(|\tilde{\mathbf{g}}|r) V_{L''}(r) j_{l'}(|\tilde{\mathbf{g}}'|r) \Big] \times C_{LL'}^{I''} \\
+ \sum_\alpha \sum_L^{n_l} \sum_{L=n_l+1}^{\infty} \sum_{L''=1}^{\tilde{n}_l} \mathrm{Q}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} \mathrm{Q}_{L',\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \Big[A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} \times \int_\alpha r^2 \mathrm{d}r u_{l,\alpha}(r) V_{L''}(r) j_{l'}(|\tilde{\mathbf{g}}'|r) \\
+ B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} \times \int_\alpha r^2 \mathrm{d}r u_{l,\alpha}(r) V_{L''}(r) j_{l'}(|\tilde{\mathbf{g}}'|r) \\
- \int_\alpha r^2 \mathrm{d}r j_l^*(|\tilde{\mathbf{g}}|r) V_{L''}(r) j_{l'}(|\tilde{\mathbf{g}}'|r) \Big] \times C_{LL'}^{L''}
\end{aligned} \tag{6 54}$$

These extra matrix elements look like a lot of extra work, and in fact they are, our new LAPW's make the matrix elements for the non-spherical potential much more complicated than for the old LAPW's To put these matrix elements into a code and still keep it efficient, we have to perform some programming that is more complex than merely transferring formulas to computer code.

6.2.2.1 Speeding up the muffin-tin matrix element calculation

Upon examination of (6 41), (6 42), and (6 54) we notice an enormous number of radial integrals To be able to calculate the Hamiltonian matrix elements, we must calculate all these integrals In this section we examine whether there is an efficient way to do this

Essentially we have two different kinds of integrals There are the integrals only containing the atomic solutions $u_{l,\alpha}(r)$ and their energy derivatives $u_{l,\alpha}(r)$ We can collect them from

(6.41), (6.42), and (6.54):

$$\begin{array}{ll} \left\langle u_{l,\alpha}\middle|H_0\middle|u_{l,\alpha}\right\rangle & \left\langle u_{l,\alpha}\middle|V\middle|u_{l,\alpha}\right\rangle \\ \left\langle u_{l,\alpha}\middle|H_0\middle|\dot{u}_{l,\alpha}\right\rangle & \left\langle \dot{u}_{l,\alpha}\middle|H_0\middle|u_{l,\alpha}\right\rangle \\ \left\langle \dot{u}_{l,\alpha}\middle|V\middle|u_{l,\alpha}\right\rangle & \left\langle \dot{u}_{l,\alpha}\middle|\dot{u}_{l,\alpha}\right\rangle \\ \left\langle \dot{u}_{l,\alpha}\middle|H_0\middle|\dot{u}_{l,\alpha}\right\rangle & \int_\alpha r^2\mathrm{d}r u^*_{l,\alpha}(r)V_{L''}(r)u_{l',\alpha}(r) \\ \int_\alpha r^2\mathrm{d}r u^*_{l,\alpha}(r)V_{L''}(r)\dot{u}_{l',\alpha}(r) & \int_\alpha r^2\mathrm{d}r \dot{u}^*_{l,\alpha}(r)V_{L''}(r)u_{l',\alpha}(r) \\ \int_\alpha r^2\mathrm{d}r \dot{u}^*_{l,\alpha}(r)V_{L''}(r)\dot{u}_{l',\alpha}(r) & \end{array} \tag{6.55}$$

These integrals need only be done for several values of l and l'. They do not depend on the basis functions through $\tilde{\mathbf{g}}$ or $\tilde{\mathbf{g}}'$.

The second kind of integral however depends on either one or both of the wave-vectors of the basis functions. These integrals are the ones that take a lot of time to calculate. So instead of performing the calculation of each integral as it is needed, we check whether we can calculate all the integrals in advance and use them as needed. This may save us from having to recalculate those integrals which occur several times because of symmetry.

We shall look at two examples where we can save a lot of time by using a symmetry argument. The first is the $\mathrm{O}^{\mathrm{PW}}_{\mathbf{g},\mathbf{g}'}$ integral as defined by (6.33).

$$\int_\alpha r^2\mathrm{d}r\ j_0(|\mathbf{g}-\mathbf{g}'|r) \tag{6.56}$$

If N is the size of the LAPW basis, this integral would be evaluated N^2 times in a straightforward implementation. But the integrand only depends on $|\mathbf{g}-\mathbf{g}'|$, so many of the integrals are actually the same. The trick is now to first find all *different* values of $|\mathbf{g}-\mathbf{g}'|$ and only calculate those integrals, which are then stored in some temporary storage with the value of $|\mathbf{g}-\mathbf{g}'|$ as index. When one of the integrals is needed, the value of $|\mathbf{g}-\mathbf{g}'|$ is calculated and is used to find the integral in temporary storage. This leads to a great saving of time when N is big.

A second example is the $\mathrm{O}^{l,\alpha}_{\tilde{\mathbf{g}},\tilde{\mathbf{g}}'}$ integral defined in (6.38a):

$$\int_\alpha r^2\mathrm{d}r\ j_l(|\tilde{\mathbf{g}}|r)j_l(|\tilde{\mathbf{g}}'|r) \tag{6.57}$$

The integral depends on both $|\tilde{\mathbf{g}}|$ and $|\tilde{\mathbf{g}}'|$. First we realise that there are actually fewer lengths of wave vectors than there are wave vectors, and secondly the following trivial symmetry holds:

$$\int_\alpha r^2\mathrm{d}r\ j_l(|\tilde{\mathbf{g}}|r)j_l(|\tilde{\mathbf{g}}'|r) = \int_\alpha r^2\mathrm{d}r\ j_l(|\tilde{\mathbf{g}}'|r)j_l(|\tilde{\mathbf{g}}|r) \tag{6.58}$$

So we first check how many lengths of $\tilde{\mathbf{g}}$ there are, how many different combinations occur, taking (6.58) into account, and we then calculate these integrals and store them.

Another way to save time is to use a less dense grid to calculate the integrals. This grid has fewer points than the grid the potential or atomic solutions are actually stored on. This saves another factor of 5 or 6 in the calculation of the integrals.

6.2.3 Warping potential terms in interstitial region

In this section I shall describe how to calculate the Hamiltonian in the interstitial region, assuming that the potential in this region is given by (see chapter 7):

$$V(\mathbf{r}) = \sum_{\mathbf{g}''} V_{\mathbf{g}''} e^{i\mathbf{g}''\,\mathbf{r}} + \sum_{\mathbf{G}''} \left\{ V^{+}_{\mathbf{G}''} e^{|\mathbf{G}''|z} + V^{-}_{\mathbf{G}''} e^{-|\mathbf{G}''|z} \right\} + V_1 z + V_2 z^2 \tag{6.59}$$

The interstitial warping potential matrix elements look like:

$$\begin{aligned} \left\langle e^{i\mathbf{g}\,\mathbf{r}} \middle| V(\mathbf{r}) \middle| e^{i\mathbf{g}'\,\mathbf{r}} \right\rangle = & \sum_{\mathbf{g}''} V_{\mathbf{g}''} \left\langle e^{i\mathbf{g}\,\mathbf{r}} \middle| e^{i\mathbf{g}''\,\mathbf{r}} \middle| e^{i\mathbf{g}'\,\mathbf{r}} \right\rangle \\ & + \sum_{\mathbf{G}''} \left\{ V^{+}_{\mathbf{G}''} \left\langle e^{i\mathbf{g}\,\mathbf{r}} \middle| e^{|\mathbf{G}''|z} \middle| e^{i\mathbf{g}'\,\mathbf{r}} \right\rangle + V^{-}_{\mathbf{G}''} \left\langle e^{i\mathbf{g}\,\mathbf{r}} \middle| e^{-|\mathbf{G}''|z} \middle| e^{i\mathbf{g}'\,\mathbf{r}} \right\rangle \right\} \\ & + V_1 \left\langle e^{i\mathbf{g}\,\mathbf{r}} \middle| z \middle| e^{i\mathbf{g}'\,\mathbf{r}} \right\rangle + V_2 \left\langle e^{i\mathbf{g}\,\mathbf{r}} \middle| z^2 \middle| e^{i\mathbf{g}'\,\mathbf{r}} \right\rangle \end{aligned} \tag{6.60}$$

When we write this out explicitly we obtain the following expression:

$$\begin{aligned} \left\langle e^{i\mathbf{g}\,\mathbf{r}} \middle| V(\mathbf{r}) \middle| e^{i\mathbf{g}'\,\mathbf{r}} \right\rangle = & \sum_{\mathbf{g}''} V_{\mathbf{g}''} \int_I \mathrm{d}\mathbf{r}^3 e^{i(\mathbf{g}'+\mathbf{g}''-\mathbf{g})\,\mathbf{r}} \\ & + \sum_{\mathbf{G}''} \left\{ V^{+}_{\mathbf{G}''} \int_I \mathrm{d}\mathbf{r}^3 e^{i(\mathbf{g}'-\mathbf{g})\,\mathbf{r}} e^{|\mathbf{G}''|z} + \right. \\ & \qquad \left. V^{-}_{\mathbf{G}''} \int_I \mathrm{d}\mathbf{r}^3 e^{i(\mathbf{g}'-\mathbf{g})\,\mathbf{r}} e^{-|\mathbf{G}''|z} \right\} \\ & + V_1 \int_I \mathrm{d}\mathbf{r}^3 z e^{i(\mathbf{g}'-\mathbf{g})\,\mathbf{r}} + V_2 \int_I \mathrm{d}\mathbf{r}^3 z^2 e^{i(\mathbf{g}'-\mathbf{g})\,\mathbf{r}} \end{aligned} \tag{6.61}$$

where I is the interstitial volume. This volume can have a rather complicated shape, so the integrals are not trivial. When we examine (6.61) we see that there are actually four different types of integrals. We shall write these now in a more general form and assign names to them:

$$\texttt{INTWP} = \int_I \mathrm{d}\mathbf{r}^3 e^{i\tilde{\mathbf{g}}\,\mathbf{r}} \tag{6.62a}$$

$$\texttt{INTWPX} = \int_I \mathrm{d}\mathbf{r}^3 e^{\tilde{G}z} e^{i\tilde{\mathbf{g}}\,\mathbf{r}} \tag{6.62b}$$

$$\texttt{INTWP1} = \int_I \mathrm{d}\mathbf{r}^3 z e^{i\tilde{\mathbf{g}}\,\mathbf{r}} \tag{6.62c}$$

$$\texttt{INTWP2} = \int_I \mathrm{d}\mathbf{r}^3 z^2 e^{i\tilde{\mathbf{g}}\,\mathbf{r}} \tag{6.62d}$$

The names assigned to the different types of integrals in (6.62) are actually the names of the subroutines that calculate them in the computer code.

Before we can evaluate these integrals we have to define what the interstitial region actually is:

> *The interstitial region is the space in a subvolume which does not belong to muffin-tins of that subvolume or neighbouring subvolumes.*

Since we usually have some sort of protrusion into a subvolume by muffin-tins from neighbouring subvolumes, the interstitial volume is a non-trivial shape. We normally evaluate the integrals over the complicated volumes in the following manner:

- construct the integral over the entire subvolume;
- subtract all parts of muffin-tins belonging to this subvolume;
- subtract all parts of muffin-tins belonging to neighbouring subvolumes.

Solving the integral over the entire subvolume is already quite complicated, even more so through the caps sticking in and out of the subvolume. In this thesis I shall not give the derivations of these integrals; they are evaluated using a mixture of analytical solutions and numerical integration.

When we count how many times these integrals are calculated, we see that we have to find a way to perform the calculation of the Hamiltonian matrix elements given in (6.61) as efficiently as possible.

6.2.3.1 Speeding up calculation of the interstitial warping matrix elements

Once we have the subroutines to evaluate the integrals in (6.62), which we call $V_{\mathbf{gg'}}$, we can implement the calculation of the matrix elements in the following manner:

```
BEGIN LOOP g
    BEGIN LOOP g'
        SET V_gg' = 0
        evaluate INTWP1 with g̃ = g' − g
        add V_1 × INTWP1 to V_gg'
        evaluate INTWP2 with g̃ = g' − g
        add V_2 × INTWP2 to V_gg'
        BEGIN LOOP G''
            evaluate INTWPX with G̃ = |G''| and g̃ = g' − g
            add V+_G'' × INTWPX to V_gg'
            evaluate INTWPX with G̃ = −|G''| and g̃ = g' − g
            add V−_G'' × INTWPX to V_gg'
            BEGIN LOOP g''_z
              SET g'' = (G'', g''_z)
              evaluate INTWP with g̃ = g' + g'' − g
              add V_g'' × INTWP to V_gg'
            END LOOP g''_z
        END LOOP G''
    END LOOP g'
END LOOP g
```

This is the logical way to implement (6.61). However, we see four nested loops, of which the outer two run over the number of LAPW basis functions, i.e. typically of the order 10^2. The third loop runs typically over about 15 to 30 $\mathbf{G}''$-vectors and the fourth loop averages about 5 to 20 g_z'' values. Some quick multiplication tells us that in a small calculation (say 80 LAPW's, 13 $\mathbf{G}$-vectors and an average of 8 g_z values) the integral `INTWP` is evaluated about 665600 times. Also the integral `INTWPX` is evaluated 166400 times. `INTWP` and `INTWPX` are non-trivial integrals, which means that the calculation of the interstitial warping potential matrix elements takes a lot of time, when done in this way.

Is there a way to calculate (6.61) more efficiently? There is, but it is not trivial. First we have to realise that the integrals with different $\tilde{\mathbf{g}}$ with the same $\tilde{\mathbf{G}}$ have much in common. Actually we can construct new subroutines which will perform all the integrals for a given $\tilde{\mathbf{G}}$ over a range of different $\widetilde{g_z}$. These new subroutines take longer than the old subroutines, but we save doing a lot of double work, so a new subroutine which does 10 different $\widetilde{g_z}$ will not take 10 times as long as the old subroutine. We will use the same name for the new subroutines, but the integrals will now get an index which labels the $\widetilde{g_z}$. The structure of the calculation of (6.61) then becomes as shown on the next page.

```
BEGIN LOOP G
    BEGIN LOOP G'
        set G̃ = G' − G
        evaluate INTWP1(g̃_z)
        evaluate INTWP2(g̃_z)
        BEGIN LOOP g_z
            BEGIN LOOP g'_z
                set g = (G, g_z)
                set g' = (G', g'_z)
                add V_1 × INTWP1(g'_z − g_z) to V_gg'
                add V_2 × INTWP2(g'_z − g_z) to V_gg'
            END LOOP g'_z
        END LOOP g_z
        BEGIN LOOP G''
            set G̃ = G' + G'' − G
            set G̃ = |G''|
            evaluate INTWPX(g̃_z, 1)
            set G̃ = −|G''|
            evaluate INTWPX(g̃_z, 2)
            set G̃ = G' + G'' − G
            evaluate INTWP(g̃_z)

            BEGIN LOOP g_z
                BEGIN LOOP g'_z
                    BEGIN LOOP g''_z
                      set g = (G, g_z)
                      set g' = (G', g'_z)
                      add V^+_G'' × INTWPX(g'_z − g_z, 1) to V_gg'
                      add V_G'' × INTWPX(g'_z − g_z, 2) to V_gg'
                      add V_g'' × INTWP(g'_z + g''_z − g_z) to V_gg'
                    END LOOP g''_z
                END LOOP g'_z
            END LOOP g_z
        END LOOP G''
    END LOOP g'
END LOOP g
```

First of all we note that all loops over g_z, g'_z, and g''_z take hardly any time, because they are just simple multiplications and additions. Secondly, the number of *calls* to the subroutines has gone down drastically. For our small system mentioned before, the new INTWPX is called only 4394 times and the new INTWP is called only 2197 times.

However, we note that the storage space has gone up, because for each call to a subroutine we have to store an array of integrals. It seems that an increase of performance in computing time will lead to an increase of the amount of storage required. The question which arises is whether we can improve matters even more. To do this we closer examine the procedure just outlined.

For example, suppose we have $N_{\mathbf{G}}$ different **G**-vectors (and thus **G**′-vectors), then INTWP1 and INTWP2 are called $(N_{\mathbf{G}})^2$ times. But not all of the $\tilde{\mathbf{G}}$ used to call the subroutines are different. Also the subroutine INTWPX is called $2(N_{\mathbf{G}})^2 N_{\mathbf{G}''}$ times, and certainly not all of the $\tilde{\mathbf{G}}$ used are different. So again it turns out we are evaluating the same integrals more than once. And the solution is again to increase the amount of intermediate storage.

The first step in our new procedure is to run a loop over all different possible combinations of **G**, **G**′, and **G**″ to list all possible $\tilde{\mathbf{G}}$.

```
set COUNTER = 1
BEGIN LOOP G
    BEGIN LOOP G'
        BEGIN LOOP G''
            set l = (G' + G'' - G)_x
            set m = (G' + G'' - G)_y
            IF [(l,m)not found before] THEN
                set ST_G̃(COUNTER) = G' + G'' - G
                set INDEX(G̃) = COUNTER
                set COUNTER = COUNTER + 1
            ENDIF

        END LOOP G''
    END LOOP G'
END LOOP G
set NR_G̃ = COUNTER -1
```

Once we have found all the different $\tilde{\mathbf{G}}$, we have to find all different $|\tilde{\mathbf{G}}|$. This is fairly simple, because all **G**-vectors in a star have the same length. A star of **G**-vectors is the set of all symmetry-related **G**-vectors. We now calculate all integrals we need only once, i.e. we run over the different $\tilde{\mathbf{G}}$-vectors and different $|\tilde{\mathbf{G}}|$, call the subroutines and store the results. Once we have those integrals, we can then construct the matrix elements.

The final loop structure then looks like:

```
BEGIN LOOP i from 1 to NR_G̃
    set G̃ =ST_G̃(i)
    evaluate INTWP(i, g_z)
    evaluate INTWP1(i, g_z)
    evaluate INTWP2(i, g_z)

    BEGIN LOOP G̃
        evaluate INTWPX(i, G̃, g_z, 1) with +G̃
        evaluate INTWPX(i, G̃, g_z, 2) with −G̃
    END LOOP G̃
END LOOP i
```

```
BEGIN LOOP G
    BEGIN LOOP G'
        set G̃ = G' − G
        set i = INDEX(G̃)
        BEGIN LOOP g_z
            BEGIN LOOP g'_z
                set g = (G, g_z)
                set g' = (G', g'_z)
                add V_1 × INTWP1(i, g'_z − g_z) to V_gg'
                add V_2 × INTWP2(i, g'_z − g_z) to V_gg'
            END LOOP g'_z
        END LOOP g_z
        BEGIN LOOP G''
            set G̃ = G' + G'' − G
            set i = INDEX(G̃)
            set G̃ = |G''|
            BEGIN LOOP g_z
                BEGIN LOOP g'_z      set g = (G, g_z)
                    set g' = (G', g'_z)
                    add V^+_G'' × INTWPX(i, G̃, g'_z − g_z, 1) to V_gg'
                    add V^−_G'' × INTWPX(i, G̃, g'_z − g_z, 2) to V_gg'
                    BEGIN LOOP g''_z
                      set g'' = (G'', g''_z)
                      add V_g'' × INTWP(i, g'_z + g''_z − g_z) to V_gg'
                    END LOOP g''_z
                END LOOP g'_z
            END LOOP g_z
        END LOOP G''
    END LOOP g'
END LOOP g
```

Using this procedure we can calculate the matrix elements defined in (6.61) a lot more efficiently than by the procedure described first. The decrease in computing time is several orders of magnitude.

6.2.4 Interstitial Hamiltonian matrix elements

In the previous section we have described how to implement the matrix elements of the interstitial warping potential. However, that is not the complete Hamiltonian in the interstitial. We shall also have to construct a kinetic energy term in the interstitial region:

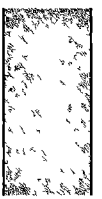
$$\left\langle \phi_{\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r}) \middle| T \middle| \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel}(\mathbf{r}) \right\rangle = \left\langle e^{\imath(\mathbf{g}+\mathbf{k}_\parallel)\cdot\mathbf{r}} \middle| \frac{1}{2m}\nabla^2 \middle| e^{\imath(\mathbf{g}'+\mathbf{k}_\parallel)\cdot\mathbf{r}} \right\rangle \tag{6.63}$$

This becomes simply:

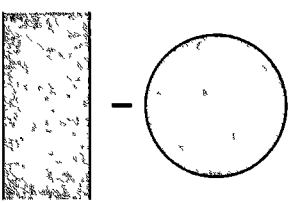
$$\left\langle \phi_{\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r}) \middle| T \middle| \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel}(\mathbf{r}) \right\rangle = \frac{1}{2m}|\mathbf{g}'+\mathbf{k}_\parallel|^2 \left\langle e^{\imath(\mathbf{g}+\mathbf{k}_\parallel)\,\mathbf{r}} \middle| e^{\imath(\mathbf{g}'+\mathbf{k}_\parallel)\cdot\mathbf{r}} \right\rangle \tag{6.64}$$

In other words the kinetic energy in the interstitial is directly related to the overlap matrix. The next question is whether we need to calculate the overlap integral in (6.64) over the complex volume of the interstitial as described in the previous section. It turns out that that is not the case. Since we actually have to shift the embedding surface from the bumpy surface, not intersecting atoms (section 2.5), we use the following procedure.

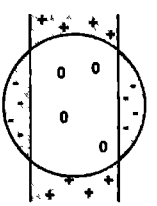
- We calculate the integral in the entire unit cell. See the following figure:



- Subtract the integral in the muffin-tin sphere, as in the following figure.



- This gives the situation as sketched in the following figure:



The integral that we are left with now contains plus the kinetic energy in the shaded region with the plusses, zero kinetic energy in the white area with the zeros, and minus the kinetic energy in the shaded area with the minuses.

When we use this procedure to construct the interstitial kinetic energy matrix elements, and therefore also the overlap matrix elements, we have essentially subtracted an Hamiltonian with zero potential in the caps. This is the procedure outlined in section 2.5 to move the embedding surface.

In this case the fact that we need to subtract the Hamiltonian with zero potential in the caps, enables us to calculate the matrix elements in the interstitial fairly simply, because the integrals in the entire unit cell and in the entire sphere can be done analytically and are therefore easy to implement and fast in computing terms.

6.2.5 Other Hamiltonian contributions

So far, we have derived expressions for Hamiltonian matrix elements, and discussed implementation of some of them. In this section I shall discuss a term that has to be added to the Hamiltonian as a correction.

We have chosen the Hamiltonian in the muffin-tin to be (semi-)relativistic, however in the interstitial we use a non-relativistic Hamiltonian. This leads to a discontinuity of the kinetic energy across the muffin-tin sphere. The kinetic energy inside the muffin-tin is of the form:

$$\left\langle \phi_{\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r}) \middle| T_{inside} \middle| \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel}(\mathbf{r}) \right\rangle = \int_{\mathrm{MT}} \mathrm{d}^3\mathbf{r} \phi_{\mathbf{g}}^{\mathbf{k}_\parallel *}(\mathbf{r}) \left(\frac{1}{2M} \nabla^2 \right) \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel}(\mathbf{r}) + \frac{1}{2M} \int_S \mathrm{d}^2\mathbf{r}_S \phi_{\mathbf{g}}^{\mathbf{k}_\parallel *}(\mathbf{r}_S) \frac{\partial}{\partial n_S} \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel}(\mathbf{r}_S) \tag{6.65}$$

The first term on the right-hand side of (6.65) is the traditional kinetic energy term, taken care of by the Hamiltonian. The second term is needed to make the Hamiltonian inside the muffin-tin hermitian. Normally this term would cancel with a similar term on the outside of the muffin-tin. However, not in our case, because the kinetic energy outside the muffin-tin looks like:

$$\left\langle \phi_{\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r}) \middle| T_{outside} \middle| \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel}(\mathbf{r}) \right\rangle = \int_{\mathrm{MT}} \mathrm{d}^3\mathbf{r} \phi_{\mathbf{g}}^{\mathbf{k}_\parallel *}(\mathbf{r}) \left(\frac{1}{2m} \nabla^2 \right) \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel}(\mathbf{r}) - \frac{1}{2m} \int_S \mathrm{d}^2\mathbf{r}_S \phi_{\mathbf{g}}^{\mathbf{k}_\parallel *}(\mathbf{r}_S) \frac{\partial}{\partial n_S} \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel}(\mathbf{r}_S) \tag{6.66}$$

Again the first term is taken care of by the interstitial Hamiltonian. However, the second term does not cancel against the second term in (6.65) because of the mass difference. So we get a correction term to the Hamiltonian, because of the mass discontinuity across the muffin-tin sphere. This term is:

$$\left\langle \phi_{\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r}) \middle| \Delta T \middle| \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel}(\mathbf{r}) \right\rangle = \tfrac{1}{2} \left(\frac{1}{M}\bigg|_{MT} - \frac{1}{m} \right) \int_S \mathrm{d}^2\mathbf{r}_S \phi_{\mathbf{g}}^{\mathbf{k}_\parallel *}(\mathbf{r}_S) \frac{\partial}{\partial n_S} \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel}(\mathbf{r}_S) \tag{6.67}$$

Using the expression for the relativistic mass:

$$M(r) = m + \frac{E - V_0(r)}{2c^2} + \mathcal{O}(\frac{1}{c^4}) \tag{6.68}$$

we obtain:

$$\frac{1}{M}\bigg|_{MT} - \frac{1}{m} = \left[m + \frac{E - V_0(r_{MT})}{2c^2} \right]^{-1} - \frac{1}{m} = -\frac{E - V_0(r_{MT})}{2m^2c^2} \tag{6.69}$$

Substituting (6.69) into (6.67) gives:

$$\left\langle \phi_{\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r}) \middle| \Delta T \middle| \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel}(\mathbf{r}) \right\rangle = -\frac{E - V_0(r_{MT})}{4m^2c^2} \int_S \mathrm{d}^2\mathbf{r}_S \phi_{\mathbf{g}}^{\mathbf{k}_\parallel *}(\mathbf{r}_S) \frac{\partial}{\partial n_S} \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel}(\mathbf{r}_S) \tag{6.70}$$

We shall evaluate this expression just outside the muffin-tin, at $r = r_{MT} + \epsilon$, so we can use the plane wave expression for the basis functions:

$$\phi_{\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r}) = e^{i(\mathbf{g}+\mathbf{k}_\parallel)\cdot\mathbf{r}} \qquad \text{and} \qquad \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel}(\mathbf{r}) = e^{i(\mathbf{g}'+\mathbf{k}_\parallel)\cdot\mathbf{r}} \tag{6.71}$$

This leads to:

$$\int_S \mathrm{d}^2\mathbf{r}_S \phi_{\mathbf{g}}^{\mathbf{k}_\parallel *}(\mathbf{r}_S) \frac{\partial}{\partial n_S} \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel}(\mathbf{r}_S) = \int_S \mathrm{d}^2\mathbf{r}_S e^{-\imath(\mathbf{g}+\mathbf{k}_\parallel)\cdot\mathbf{r}} \frac{\partial}{\partial n_S} e^{\imath(\mathbf{g}'+\mathbf{k}_\parallel)\,\mathbf{r}} \tag{6.72}$$

We can choose our z-axis to be parallel to $\mathbf{g}' + \mathbf{k}_\parallel$, to obtain:

$$\begin{aligned}
&\int_S \mathrm{d}^2\mathbf{r}_S e^{-\imath(\mathbf{g}+\mathbf{k}_\parallel)\,\mathbf{r}} \frac{\partial}{\partial r} e^{\imath|\mathbf{g}'+\mathbf{k}_\parallel| r\cos\theta} \\
&\quad = i|\mathbf{g}'+\mathbf{k}_\parallel| \int_S \mathrm{d}^2\mathbf{r}_S e^{-\imath(\mathbf{g}+\mathbf{k}_\parallel)\,\mathbf{r}} e^{\imath(\mathbf{g}'+\mathbf{k}_\parallel)\cdot\mathbf{r}} \cos\theta \\
&\quad = i|\mathbf{g}'+\mathbf{k}_\parallel| r_{MT}^2 \int_S \mathrm{d}\hat{r} e^{\imath(\mathbf{g}'-\mathbf{g})\,\mathbf{r}} \cos\theta \\
&\quad = i|\mathbf{g}'+\mathbf{k}_\parallel| r_{MT}^2 \int_S \mathrm{d}\hat{r} \sum_L^\infty 4\pi i^l j_l(|\mathbf{g}'-\mathbf{g}| r_{MT}) \times \\
&\qquad\qquad Y_L^*(\widehat{\mathbf{g}'-\mathbf{g}}) Y_L(\hat{r}) \cos\theta \\
&\quad = 4\pi i|\mathbf{g}'+\mathbf{k}_\parallel| r_{MT}^2 \sum_L^\infty i^l j_l(|\mathbf{g}'-\mathbf{g}| r_{MT}) Y_L^*(\widehat{\mathbf{g}'-\mathbf{g}}) \times \\
&\qquad\qquad \int_S \mathrm{d}\hat{r} Y_L(\hat{r}) 2\sqrt{\frac{\pi}{3}} Y_{10}(\hat{r}) \\
&\quad = 4\pi i|\mathbf{g}'+\mathbf{k}_\parallel| r_{MT}^2 \sum_L^\infty i^l j_l(|\mathbf{g}'-\mathbf{g}| r_{MT}) Y_L^*(\widehat{\mathbf{g}'-\mathbf{g}}) 2\sqrt{\frac{\pi}{3}} \delta_{L,(1,0)} \\
&\quad = 4\pi i|\mathbf{g}'+\mathbf{k}_\parallel| r_{MT}^2 \imath j_1(|\mathbf{g}'-\mathbf{g}| r_{MT}) \frac{1}{2}\sqrt{\frac{3}{\pi}} \cos(\widehat{\mathbf{g}'-\mathbf{g}}) 2\sqrt{\frac{\pi}{3}} \\
&\quad = -4\pi|\mathbf{g}'+\mathbf{k}_\parallel| r_{MT}^2 j_1(|\mathbf{g}'-\mathbf{g}| r_{MT}) \frac{(\mathbf{g}'-\mathbf{g})\cdot\mathbf{g}'}{|\mathbf{g}'-\mathbf{g}||\mathbf{g}'|}
\end{aligned} \tag{6.73}$$

This means that the correction term looks like:

$$\left\langle \phi_{\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r}) \middle| \Delta T \middle| \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel}(\mathbf{r}) \right\rangle = \frac{E - V_0(r_{MT})}{m^2c^2} \pi|\mathbf{g}'+\mathbf{k}_\parallel| r_{MT}^2 j_1(|\mathbf{g}'-\mathbf{g}| r_{MT}) \frac{(\mathbf{g}'-\mathbf{g})\cdot\mathbf{g}'}{|\mathbf{g}'-\mathbf{g}||\mathbf{g}'|} \tag{6.74}$$

This is the last of the contributions to the Hamiltonian which arises from quantum mechanics. In section 6.2.7 we shall discuss some Hamiltonian matrix elements arising from embedding theory. Before that we shall briefly discuss the overlap matrix elements. The overlap matrix comes into the problem because our LAPW basis is not orthonormal.

6.2.6 Overlap matrix elements

The interstitial overlap has already been mentioned in the section on the interstitial Hamiltonian, so here we will only examine the overlap in the muffin-tin. The muffin-tin overlap matrix elements contain many of the same elements as the muffin-tin Hamiltonian. The

general expression is:

$$\left\langle \phi_{\mathbf{g}}^{\mathbf{k}_\parallel} \middle| \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel} \right\rangle = \left\langle e^{i\mathbf{g}\cdot\mathbf{r}} \middle| e^{i\mathbf{g}'\cdot\mathbf{r}} \right\rangle + \sum_\alpha \sum_L^{n_l} \mathrm{Q}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} \mathrm{Q}_{L,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \tag{6.75}$$

$$\begin{aligned} &\Big[A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} A_{l,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \cdot \int_\alpha r^2 \mathrm{d}r u_{l,\alpha}^*(r) u_{l,\alpha}(r) \\ &+ A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} B_{l,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \cdot \int_\alpha r^2 \mathrm{d}r u_{l,\alpha}^*(r) \dot{u}_{l,\alpha}(r) \\ &+ B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} A_{l,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \cdot \int_\alpha r^2 \mathrm{d}r \dot{u}_{l,\alpha}^*(r) u_{l,\alpha}(r) \\ &+ B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} B_{l,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \cdot \int_\alpha r^2 \mathrm{d}r \dot{u}_{l,\alpha}^*(r) \dot{u}_{l,\alpha}(r) \\ &- \int_\alpha r^2 \mathrm{d}r j_l^*(|\mathbf{g}+\mathbf{k}_\parallel| r) j_l(|\mathbf{g}'+\mathbf{k}_\parallel| r) \Big] \end{aligned} \tag{6.76}$$

This expression is very similar to (6.26). Using the integrals derived for the Hamiltonian, we obtain the following expression for (6.76):

$$\begin{aligned} \mathrm{O}_{\mathbf{g}\mathbf{g}'} = \mathrm{O}_{\mathbf{g},\mathbf{g}'}^{\mathrm{PW}} + \sum_\alpha \sum_L^{n_l} \mathrm{Q}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} \mathrm{Q}_{L,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \Big[& A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} A_{l,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \langle u_{l,\alpha} | u_{l,\alpha} \rangle \\ & + B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel *} B_{l,\alpha}^{\mathbf{g}',\mathbf{k}_\parallel} \times \langle \dot{u}_{l,\alpha} | \dot{u}_{l,\alpha} \rangle \\ & - \mathrm{O}_{\mathbf{g},\mathbf{g}'}^{l,\alpha} \Big] \end{aligned} \tag{6.77}$$

6.2.7 Matrix elements related to embedding

As shown in the basic embedding equation, (2.10), which reads:

$$\left[-\tfrac{1}{2}\nabla_{\mathbf{r}}^2 + V'(\mathbf{r}) + \tfrac{1}{2}\frac{\partial}{\partial n_S}\delta(\mathbf{r}-\mathbf{r}_S) + \widehat{G}_0^{-1}(E)\delta(\mathbf{r}-\mathbf{r}_S)\delta(\mathbf{r}'-\mathbf{r}_S') \right] \phi_1(\mathbf{r}) = E\phi_1(\mathbf{r}) \tag{6.78}$$

there are some new surface terms in the Hamiltonian in embedding theory, namely the normal derivative term and the embedding potential on the embedding surface.

The normal derivative term gives the following matrix element:

$$\begin{aligned} \left\langle \phi_{\mathbf{g}}^{\mathbf{k}_\parallel} \middle| \tfrac{1}{2}\frac{\partial}{\partial n_S}\delta(\mathbf{r}-\mathbf{r}_S) \middle| \phi_{\mathbf{g}'}^{\mathbf{k}_\parallel} \right\rangle &= \tfrac{1}{2}\int_V \mathrm{d}^3\mathbf{r} e^{-i(\mathbf{g}+\mathbf{k}_\parallel)\cdot\mathbf{r}} \frac{\partial}{\partial z} e^{i(\mathbf{g}'+\mathbf{k}_\parallel)\cdot\mathbf{r}} \delta(\mathbf{r}-\mathbf{r}_S) \\ &= \pm \tfrac{1}{2} g_z' e^{i(g_z'-g_z)z_S} \delta_{\mathbf{G},\mathbf{G}'} \end{aligned} \tag{6.79}$$

where I have used the fact that in my implementation of the embedding method the embedding planes are always perpendicular to the z-direction. The sign of (6.79) depends on whether we deal with a right or left embedding plane. For a left-hand embedding plane, the normal derivative points into the system we are embedding in, i.e. to the left, which means a minus sign in (6.79). For a right-hand embedding plane, the normal derivative points to the right and we get a plus sign.

For the embedding potential term, we use the fact that the embedding potential is expanded in terms of the planar basis defined in (3.95):

$$e^{\mathbf{k}_\parallel}_{\tilde{\mathbf{G}}}(\mathbf{R}_S) = \frac{1}{\sqrt{A}} e^{\imath\left(\mathbf{k}_\parallel + \tilde{\mathbf{G}}\right)\mathbf{R}_S} \tag{6.80}$$

and can be written as:

$$\widehat{G_0}^{-1}(\mathbf{R}_S, \mathbf{R}'_S; \mathcal{E}, \mathbf{k}_\parallel) = \sum_{\tilde{\mathbf{G}}\tilde{\mathbf{G}}'} e^{\mathbf{k}_\parallel}_{\tilde{\mathbf{G}}}(\mathbf{R}_S) \left(\widehat{G_0}^{-1}(\mathcal{E}, \mathbf{k}_\parallel)\right)_{\tilde{\mathbf{G}}\tilde{\mathbf{G}}'} e^{\mathbf{k}_\parallel\,*}_{\tilde{\mathbf{G}}'}(\mathbf{R}'_S) \tag{6.81}$$

This means that the matrix elements of the embedding potential follow from:

$$\begin{aligned}
&\int_V \mathrm{d}^3\mathbf{r} \int_V \mathrm{d}^3\mathbf{r}' e^{-\imath(\mathbf{g}+\mathbf{k}_\parallel)\,\mathbf{r}} \sum_{\tilde{\mathbf{G}}\tilde{\mathbf{G}}'} e^{\mathbf{k}_\parallel}_{\tilde{\mathbf{G}}}(\mathbf{r}) \left(\widehat{G_0}^{-1}(\mathcal{E}, \mathbf{k}_\parallel)\right)_{\tilde{\mathbf{G}}\tilde{\mathbf{G}}'} \times \\
&\qquad\qquad e^{\mathbf{k}_\parallel\,*}_{\tilde{\mathbf{G}}'}(\mathbf{r}') e^{\imath(\mathbf{g}'+\mathbf{k}_\parallel)\cdot\mathbf{r}} \delta(\mathbf{r} - \mathbf{R}_S)\delta(\mathbf{r} - \mathbf{R}'_S) \\
&= \sum_{\tilde{\mathbf{G}}\tilde{\mathbf{G}}'} \int_S \mathrm{d}^2\mathbf{R}_S e^{-\imath(\mathbf{g}+\mathbf{k}_\parallel)\cdot\mathbf{R}_S} e^{\mathbf{k}_\parallel}_{\tilde{\mathbf{G}}}(\mathbf{R}_S) \left(\widehat{G_0}^{-1}(\mathcal{E}, \mathbf{k}_\parallel)\right)_{\tilde{\mathbf{G}}\tilde{\mathbf{G}}'} \times \\
&\qquad\qquad \int_S \mathrm{d}^2\mathbf{R}'_S e^{\mathbf{k}_\parallel\,*}_{\tilde{\mathbf{G}}'}(\mathbf{R}'_S) e^{\imath(\mathbf{g}'+\mathbf{k}_\parallel)\cdot\mathbf{R}'_S}
\end{aligned} \tag{6.82}$$

So we need to calculate the overlap between an interstitial LAPW and the planar basis function:

$$\begin{aligned}
\int_S \mathrm{d}^2\mathbf{R}_S e^{-\imath(\mathbf{g}+\mathbf{k}_\parallel)\cdot\mathbf{R}_S} e^{\mathbf{k}_\parallel}_{\tilde{\mathbf{G}}}(\mathbf{R}_S) &= \int_S \mathrm{d}^2\mathbf{R}_S e^{-i(\mathbf{G}+\mathbf{k}_\parallel)\,\mathbf{R}_S} e^{-\imath g_z z_S} \frac{1}{\sqrt{A}} e^{\imath\left(\mathbf{k}_\parallel+\tilde{\mathbf{G}}\right)\cdot\mathbf{R}_S} \\
&= \frac{1}{\sqrt{A}} e^{-\imath g_z z_S} \int_S \mathrm{d}^2\mathbf{R}_S e^{\imath(\tilde{\mathbf{G}}-\mathbf{G})\,\mathbf{R}_S} \\
&= \frac{1}{\sqrt{A}} e^{-\imath g_z z_S} \delta_{\tilde{\mathbf{G}},\mathbf{G}} A \\
&= \sqrt{A} e^{-\imath g_z z_S} \delta_{\tilde{\mathbf{G}},\mathbf{G}} \\
&= C^{S\,\dagger}_{\mathbf{g}\mathbf{G}}
\end{aligned} \tag{6.83}$$

so that the matrix elements of the embedding potential in terms of the LAPW basis become:

$$\begin{aligned}
\left(\widehat{G_0}^{-1}(\mathcal{E}, \mathbf{k}_\parallel)\right)_{\mathbf{g}\mathbf{g}'} &= C^{S\,\dagger}_{\mathbf{g}\tilde{\mathbf{G}}} \left(\widehat{G_0}^{-1}(\mathcal{E}, \mathbf{k}_\parallel)\right)_{\tilde{\mathbf{G}}\tilde{\mathbf{G}}'} C^S_{\tilde{\mathbf{G}}'\mathbf{g}'} \\
&= \sqrt{A} e^{-\imath g_z z_S} \left(\widehat{G_0}^{-1}(\mathcal{E}, \mathbf{k}_\parallel)\right)_{\tilde{\mathbf{G}}\tilde{\mathbf{G}}'} \sqrt{A} e^{\imath g'_z z_S} \\
&= A e^{\imath(g'_z - g_z) z_S} \left(\widehat{G_0}^{-1}(\mathcal{E}, \mathbf{k}_\parallel)\right)_{\tilde{\mathbf{G}}\tilde{\mathbf{G}}'}
\end{aligned} \tag{6.84}$$

where S indicates the embedding surface, i.e. whether we have a right-hand or left-hand embedding potential.

We now have all the contributions to the Hamiltonian matrix. The next step is to construct the Green function matrix.

6.3 Constructing and using Green functions

As we have already discussed in section 1.2.5 and seen in (1.23), the Green function is the operator inverse of the Hamiltonian minus the energy times the overlap.

$$G(\mathcal{E}, \mathbf{k}_{\|}) = \left[H(\mathcal{E}, \mathbf{k}_{\|}) - \mathcal{E}O(\mathbf{k}_{\|})\right]^{-1} \tag{6.85}$$

We can replace the operators with matrices, and (6.85) becomes a matrix equation. So once we have the Hamiltonian and overlap matrices, calculating the Green function matrix becomes a process of matrix inversion.

In the GROW module of the embedding code, this is done in the following way:

1. A $\mathbf{k}_{\|}$ is chosen.
2. For that $\mathbf{k}_{\|}$ the energy independent Hamiltonian in the muffin-tin and in the interstitial + all correction terms + normal derivative terms on the embedding planes are stored in the matrix H_1.
3. The part of the Hamiltonian linear in energy is stored in matrix H_2.
4. The overlap is calculated and stored in O.
5. The projectors used in (6.84) are calculated. These projectors are:

$$C^{S}_{\tilde{\mathbf{G}}\mathbf{g}} = \sqrt{A} e^{ig_z z_S} \delta_{\tilde{\mathbf{G}},\mathbf{G}} \tag{6.86}$$

   where $\mathbf{G}$ is the parallel component of $\mathbf{g}$.

6. The embedding potentials, expanded in the planar basis are projected onto the LAPW basis:

$$\left(\widehat{G_0}^{-1}(\mathcal{E}, \mathbf{k}_{\|})\right)^{L}_{\mathbf{g}\mathbf{g}'} = C^{L}_{\mathbf{g}\tilde{\mathbf{G}}}{}^{\dagger} \left(\widehat{G}^{-1}_{0,L}(\mathcal{E}, \mathbf{k}_{\|})\right)_{\tilde{\mathbf{G}}\tilde{\mathbf{G}}'} C^{L}_{\tilde{\mathbf{G}}'\mathbf{g}'} \tag{6.87a}$$

$$\left(\widehat{G_0}^{-1}(\mathcal{E}, \mathbf{k}_{\|})\right)^{R}_{\mathbf{g}\mathbf{g}'} = C^{R}_{\mathbf{g}\tilde{\mathbf{G}}}{}^{\dagger} \left(\widehat{G}^{-1}_{0,R}(\mathcal{E}, \mathbf{k}_{\|})\right)_{\tilde{\mathbf{G}}\tilde{\mathbf{G}}'} C^{R}_{\tilde{\mathbf{G}}'\mathbf{g}'} \tag{6.87b}$$

7. In the routine which calculates the Green function, these ingredients are combined according to a requested mode.

When we want the Green function, the routine will calculate the following:

$$G(\mathcal{E}, \mathbf{k}_{\|}) = \left[H_1 + \mathcal{E}H_2 + \left(\widehat{G_0}^{-1}(\mathcal{E}, \mathbf{k}_{\|})\right)^{L}_{\mathbf{g}\mathbf{g}'} + \left(\widehat{G_0}^{-1}(\mathcal{E}, \mathbf{k}_{\|})\right)^{R}_{\mathbf{g}\mathbf{g}'} - \mathcal{E}O\right]^{-1} \tag{6.88}$$

However, sometimes we want a Green function with zero normal derivative on either or both of the embedding planes. To be able to get those, the subroutine which calculates Green functions (called GETG) has an input parameter that enables it to set either the left or right embedding potential (or both) to zero. The option to set both embedding potentials to zero is used to generate the Green function for finding the scattering properties of subvolumes.

The Green function is a very handy quantity to have, as seen in section 1.2.5, because from it we can calculate physical quantities we are interested in. This will be the focus of the following sections.

6.3.1 Getting at the density of states hiding in the Green function

In chapter 1 we have seen how we can get the local density of states (energy dependent electron density) from the Green function;

$$\sigma(\mathbf{r};E,\mathbf{k}_{\|}) = \lim_{\epsilon\to 0}\frac{1}{\pi}\mathrm{Im}G(\mathbf{r},\mathbf{r};E+i\epsilon,\mathbf{k}_{\|}) \tag{6.89}$$

We normally do not take the limit of ϵ to zero, but set it very small and positive.

A useful quantity for comparison with experiment is the local density of states integrated over some volume V:

$$\sigma(\mathcal{E},\mathbf{k}_{\|}) = \int_V \mathrm{d}^3\mathbf{r}\sigma(\mathbf{r};\mathcal{E},\mathbf{k}_{\|}) = \frac{1}{\pi}\mathrm{Im}\int_V \mathrm{d}^3\mathbf{r}G(\mathbf{r},\mathbf{r};\mathcal{E},\mathbf{k}_{\|}) \tag{6.90}$$

When we substitute the matrix expression of the Green function, we find:

$$\begin{aligned}\sigma(\mathcal{E},\mathbf{k}_{\|}) =& \frac{1}{\pi}\mathrm{Im}\int_V \mathrm{d}^3\mathbf{r}\sum_{\mathbf{g}\mathbf{g}'}\phi_{\mathbf{g}}^{\mathbf{k}_{\|}}(\mathbf{r})G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_{\|})\phi_{\mathbf{g}'}^{\mathbf{k}_{\|}*}(\mathbf{r})\\ =& \frac{1}{\pi}\mathrm{Im}\sum_{\mathbf{g}\mathbf{g}'}G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_{\|})\int_V \mathrm{d}^3\mathbf{r}\phi_{\mathbf{g}}^{\mathbf{k}_{\|}}(\mathbf{r})\phi_{\mathbf{g}'}^{\mathbf{k}_{\|}*}(\mathbf{r})\\ =& \frac{1}{\pi}\mathrm{Im}\sum_{\mathbf{g}\mathbf{g}'}G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_{\|})O_{\mathbf{g}\mathbf{g}'}^{\mathbf{k}_{\|},V}\end{aligned} \tag{6.91}$$

So the density of states is fairly easy to calculate by multiplying Green function matrix elements with overlap matrix elements. Another nice result is, that if we want a density of states only in one muffin-tin, we multiply the Green function matrix elements with the matrix elements of the overlap in only that muffin-tin. The same goes for any volume: all we have to do is calculate the overlap over that volume and this gives us the density of states in that volume.

6.3.1.1 Angular momentum-resolved density of states

For some applications it is useful to be able to calculate the density of states (DOS) projected onto the different angular momenta. In this way we have an s-DOS, a p-DOS, a d-DOS, etc. As stated before, what we have to do is to calculate the overlap in a certain region, but now also projected onto a certain angular momentum value. When we examine the expression for the muffin-tin overlap, (6.77), we notice that we can write:

$$\mathrm{O}_{\mathbf{g}\mathbf{g}'}^{\mathbf{k}_{\|},l,\alpha} = \mathrm{Q}_{L,\alpha}^{\mathbf{g},\mathbf{k}_{\|}*}\mathrm{Q}_{L,\alpha}^{\mathbf{g}',\mathbf{k}_{\|}}\left[A_{l,\alpha}^{\mathbf{g},\mathbf{k}_{\|}*}A_{l,\alpha}^{\mathbf{g}',\mathbf{k}_{\|}}\left\langle u_{l,\alpha}\middle|u_{l,\alpha}\right\rangle + B_{l,\alpha}^{\mathbf{g},\mathbf{k}_{\|}*}B_{l,\alpha}^{\mathbf{g}',\mathbf{k}_{\|}}\left\langle \dot{u}_{l,\alpha}\middle|\dot{u}_{l,\alpha}\right\rangle\right] \tag{6.92}$$

Using this overlap in (6.91) we get the l-resolved DOS or lDOS in muffin-tin α:

$$\sigma_{l,\alpha}(\mathcal{E},\mathbf{k}_{\|}) = \frac{1}{\pi}\mathrm{Im}\sum_{\mathbf{g}\mathbf{g}'}G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_{\|})O_{\mathbf{g}\mathbf{g}'}^{\mathbf{k}_{\|},l,\alpha} \tag{6.93}$$

This DOS only has meaning in our method for those values of l for which we use augmentation in the LAPW, because for higher l's we would have to resolve the plane-wave in the muffin-tin into angular momentum components.

6.3.1.2 The (l,m)-resolved density of states

The density of states can be further resolved into m contributions as well as l. The relevant overlap is found using a similar expression to (6.92), unravelling the different l and m values. In the following expressions I have dropped the α's to keep them more readable. Also we have used the fact that the atomic solutions are normalised and:

$$\left\langle \dot{u}_l \middle| \dot{u}_l \right\rangle = N_l \tag{6.94}$$

We use the representation of (l,m) that corresponds to atomic orbitals as known from quantum mechanics. Using all this we obtain the following expressions for the (l,m) resolved overlap:

$$O_{\mathbf{gg'}}^{\mathbf{k}_\parallel,s} = Q_{(0,0)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(0,0)}^{\mathbf{g'},\mathbf{k}_\parallel} \left[A_0^{\mathbf{g},\mathbf{k}_\parallel *} A_0^{\mathbf{g'},\mathbf{k}_\parallel} + B_0^{\mathbf{g},\mathbf{k}_\parallel *} B_0^{\mathbf{g'},\mathbf{k}_\parallel} N_0 \right] \tag{6.95a}$$

$$O_{\mathbf{gg'}}^{\mathbf{k}_\parallel,p_x} = \left\{ Q_{(1,1)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(1,1)}^{\mathbf{g'},\mathbf{k}_\parallel} + Q_{(1,1)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(1,-1)}^{\mathbf{g'},\mathbf{k}_\parallel} + Q_{(1,-1)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(1,1)}^{\mathbf{g'},\mathbf{k}_\parallel} + Q_{(1,-1)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(1,-1)}^{\mathbf{g'},\mathbf{k}_\parallel} \right\} \times \left[A_1^{\mathbf{g},\mathbf{k}_\parallel *} A_1^{\mathbf{g'},\mathbf{k}_\parallel} + B_1^{\mathbf{g},\mathbf{k}_\parallel *} B_1^{\mathbf{g'},\mathbf{k}_\parallel} N_1 \right] \tag{6.95b}$$

$$O_{\mathbf{gg'}}^{\mathbf{k}_\parallel,p_y} = \left\{ Q_{(1,1)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(1,1)}^{\mathbf{g'},\mathbf{k}_\parallel} - Q_{(1,1)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(1,-1)}^{\mathbf{g'},\mathbf{k}_\parallel} - Q_{(1,-1)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(1,1)}^{\mathbf{g'},\mathbf{k}_\parallel} + Q_{(1,-1)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(1,-1)}^{\mathbf{g'},\mathbf{k}_\parallel} \right\} \times \left[A_1^{\mathbf{g},\mathbf{k}_\parallel *} A_1^{\mathbf{g'},\mathbf{k}_\parallel} + B_1^{\mathbf{g},\mathbf{k}_\parallel *} B_1^{\mathbf{g'},\mathbf{k}_\parallel} N_1 \right] \tag{6.95c}$$

$$O_{\mathbf{gg'}}^{\mathbf{k}_\parallel,p_z} = Q_{(1,0)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(1,0)}^{\mathbf{g'},\mathbf{k}_\parallel} \left[A_1^{\mathbf{g},\mathbf{k}_\parallel *} A_1^{\mathbf{g'},\mathbf{k}_\parallel} + B_1^{\mathbf{g},\mathbf{k}_\parallel *} B_1^{\mathbf{g'},\mathbf{k}_\parallel} N_1 \right] \tag{6.95d}$$

$$O_{\mathbf{gg'}}^{\mathbf{k}_\parallel,d_{xy}} = \left\{ Q_{(2,2)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(2,2)}^{\mathbf{g'},\mathbf{k}_\parallel} - Q_{(2,2)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(2,-2)}^{\mathbf{g'},\mathbf{k}_\parallel} - Q_{(2,-2)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(2,2)}^{\mathbf{g'},\mathbf{k}_\parallel} + Q_{(2,-2)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(2,-2)}^{\mathbf{g'},\mathbf{k}_\parallel} \right\} \times \left[A_2^{\mathbf{g},\mathbf{k}_\parallel *} A_2^{\mathbf{g'},\mathbf{k}_\parallel} + B_2^{\mathbf{g},\mathbf{k}_\parallel *} B_2^{\mathbf{g'},\mathbf{k}_\parallel} N_2 \right] \tag{6.95e}$$

$$O_{\mathbf{gg'}}^{\mathbf{k}_\parallel,d_{xz}} = \left\{ Q_{(2,1)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(2,1)}^{\mathbf{g'},\mathbf{k}_\parallel} - Q_{(2,1)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(2,-1)}^{\mathbf{g'},\mathbf{k}_\parallel} - Q_{(2,-1)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(2,1)}^{\mathbf{g'},\mathbf{k}_\parallel} + Q_{(2,-1)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(2,-1)}^{\mathbf{g'},\mathbf{k}_\parallel} \right\} \times \left[A_2^{\mathbf{g},\mathbf{k}_\parallel *} A_2^{\mathbf{g'},\mathbf{k}_\parallel} + B_2^{\mathbf{g},\mathbf{k}_\parallel *} B_2^{\mathbf{g'},\mathbf{k}_\parallel} N_2 \right] \tag{6.95f}$$

$$O_{\mathbf{gg'}}^{\mathbf{k}_\parallel,d_{yz}} = \left\{ Q_{(2,1)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(2,1)}^{\mathbf{g'},\mathbf{k}_\parallel} + Q_{(2,1)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(2,-1)}^{\mathbf{g'},\mathbf{k}_\parallel} + Q_{(2,-1)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(2,1)}^{\mathbf{g'},\mathbf{k}_\parallel} + Q_{(2,-1)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(2,-1)}^{\mathbf{g'},\mathbf{k}_\parallel} \right\} \times \left[A_2^{\mathbf{g},\mathbf{k}_\parallel *} A_2^{\mathbf{g'},\mathbf{k}_\parallel} + B_2^{\mathbf{g},\mathbf{k}_\parallel *} B_2^{\mathbf{g'},\mathbf{k}_\parallel} N_2 \right] \tag{6.95g}$$

$$O_{\mathbf{gg'}}^{\mathbf{k}_\parallel,d_{x^2-y^2}} = \left\{ Q_{(2,2)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(2,2)}^{\mathbf{g'},\mathbf{k}_\parallel} + Q_{(2,2)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(2,-2)}^{\mathbf{g'},\mathbf{k}_\parallel} + Q_{(2,-2)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(2,2)}^{\mathbf{g'},\mathbf{k}_\parallel} + Q_{(2,-2)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(2,-2)}^{\mathbf{g'},\mathbf{k}_\parallel} \right\} \times$$

$$\left[A_2^{\mathbf{g},\mathbf{k}_\parallel *} A_2^{\mathbf{g}',\mathbf{k}_\parallel} + B_2^{\mathbf{g},\mathbf{k}_\parallel *} B_2^{\mathbf{g}',\mathbf{k}_\parallel} N_2\right] \tag{6.95h}$$

$$O_{\mathbf{g}\mathbf{g}'}^{\mathbf{k}_\parallel, d_{z^2}} = Q_{(2,0)}^{\mathbf{g},\mathbf{k}_\parallel *} Q_{(2,0)}^{\mathbf{g}',\mathbf{k}_\parallel} \left[A_2^{\mathbf{g},\mathbf{k}_\parallel *} A_2^{\mathbf{g}',\mathbf{k}_\parallel} + B_2^{\mathbf{g},\mathbf{k}_\parallel *} B_2^{\mathbf{g}',\mathbf{k}_\parallel} N_2\right] \tag{6.95i}$$

Putting these overlaps in an expression like (6.91) gives us the DOS resolved into (l, m) contributions. This has only been implemented up to d-states.

6.3.1.3 Symmetry-resolved density of states

When comparing calculations with photoemission experiments[15], it is sometimes useful to be able to split the density of states into even and odd parts with respect to some mirror symmetry, $\mathcal{M}$, in a plane perpendicular to the atomic layers or surface. Suppose the symmetry acts on the basis functions in the following way:

$$\mathcal{M}\left(\phi_{\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r})\right) = \phi_{\mathcal{M}(\mathbf{g})}^{\mathbf{k}_\parallel}(\mathbf{r}) \tag{6.96}$$

So a basis function will always be mapped onto another basis function. We can now construct the even and odd combinations of basis functions:

$$\Psi_{even,\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r}) = \frac{1}{2}\left(\phi_{\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r}) + \mathcal{M}\left(\phi_{\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r})\right)\right) \tag{6.97a}$$

$$\Psi_{odd,\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r}) = \frac{1}{2}\left(\phi_{\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r}) - \mathcal{M}\left(\phi_{\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r})\right)\right) \tag{6.97b}$$

for which the following relationships hold:

$$\mathcal{M}\left(\Psi_{even,\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r})\right) = +\Psi_{even,\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r}) \tag{6.98a}$$

$$\mathcal{M}\left(\Psi_{odd,\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r})\right) = -\Psi_{odd,\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r}) \tag{6.98b}$$

where we have assumed a normal mirror symmetry, i.e. $\mathcal{M}^2 = I$ (we do not consider glide-planes). Similarly we can construct an even and an odd density of states:

$$\sigma_{even}(\mathcal{E}, \mathbf{k}_\parallel) = \frac{1}{2}\left(\sigma(\mathcal{E}, \mathbf{k}_\parallel) + \mathcal{M}\sigma(\mathcal{E}, \mathbf{k}_\parallel)\right) \tag{6.99a}$$

$$\sigma_{odd}(\mathcal{E}, \mathbf{k}_\parallel) = \frac{1}{2}\left(\sigma(\mathcal{E}, \mathbf{k}_\parallel) - \mathcal{M}\sigma(\mathcal{E}, \mathbf{k}_\parallel)\right) \tag{6.99b}$$

Using (6.91) we write for the even DOS:

$$\begin{aligned}
\sigma_{even}(\mathcal{E},\mathbf{k}_{\|}) =& \frac{1}{2}\frac{1}{\pi}\Big(\mathrm{Im}\sum_{\mathbf{g}\mathbf{g}'} G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_{\|}) \int_V \mathrm{d}^3\mathbf{r}\phi_{\mathbf{g}}^{\mathbf{k}_{\|}}(\mathbf{r})\phi_{\mathbf{g}'}^{\mathbf{k}_{\|}\,*}(\mathbf{r}) + \\
& \qquad \mathrm{Im}\sum_{\mathbf{g}\mathbf{g}'} G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_{\|}) \int_V \mathrm{d}^3\mathbf{r}\mathcal{M}\Big(\phi_{\mathbf{g}}^{\mathbf{k}_{\|}}(\mathbf{r})\Big)\phi_{\mathbf{g}'}^{\mathbf{k}_{\|}\,*}(\mathbf{r})\Big) \\
=& \frac{1}{2}\frac{1}{\pi}\mathrm{Im}\sum_{\mathbf{g}\mathbf{g}'}\Big(G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_{\|}) \int_V \mathrm{d}^3\mathbf{r}\phi_{\mathbf{g}}^{\mathbf{k}_{\|}}(\mathbf{r})\phi_{\mathbf{g}'}^{\mathbf{k}_{\|}\,*}(\mathbf{r}) + \\
& \qquad G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_{\|}) \int_V \mathrm{d}^3\mathbf{r}\phi_{\mathcal{M}(\mathbf{g})}^{\mathbf{k}_{\|}}(\mathbf{r})\phi_{\mathbf{g}'}^{\mathbf{k}_{\|}\,*}(\mathbf{r})\Big) \\
=& \frac{1}{2}\frac{1}{\pi}\mathrm{Im}\sum_{\mathbf{g}\mathbf{g}'}\Big(G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_{\|})O_{\mathbf{g}\mathbf{g}'}^{\mathbf{k}_{\|},V} + \\
& \qquad G_{\mathcal{M}(\mathbf{g})\mathbf{g}'}(\mathcal{E},\mathbf{k}_{\|}) \int_V \mathrm{d}^3\mathbf{r}\phi_{\mathbf{g}}^{\mathbf{k}_{\|}}(\mathbf{r})\phi_{\mathbf{g}'}^{\mathbf{k}_{\|}\,*}(\mathbf{r})\Big) \\
=& \frac{1}{2}\frac{1}{\pi}\mathrm{Im}\sum_{\mathbf{g}\mathbf{g}'}\Big(G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_{\|})O_{\mathbf{g}\mathbf{g}'}^{\mathbf{k}_{\|},V} + G_{\mathcal{M}(\mathbf{g})\mathbf{g}'}(\mathcal{E},\mathbf{k}_{\|})O_{\mathbf{g}\mathbf{g}'}^{\mathbf{k}_{\|},V}\Big) \\
=& \frac{1}{\pi}\mathrm{Im}\sum_{\mathbf{g}\mathbf{g}'}\Big\{\frac{1}{2}\Big(G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_{\|}) + G_{\mathcal{M}(\mathbf{g})\mathbf{g}'}(\mathcal{E},\mathbf{k}_{\|})\Big)\Big\}O_{\mathbf{g}\mathbf{g}'}^{\mathbf{k}_{\|},V}
\end{aligned} \tag{6.100}$$

where we have used the fact that $\mathcal{M}^{-1} = \mathcal{M}$ and the set $\{\mathcal{M}(\mathbf{g})\}$ is the same set as $\{\mathbf{g}\}$. For the odd density of states we obtain similarly:

$$\sigma_{odd}(\mathcal{E},\mathbf{k}_{\|}) = \frac{1}{\pi}\mathrm{Im}\sum_{\mathbf{g}\mathbf{g}'}\Big\{\frac{1}{2}\Big(G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_{\|}) - G_{\mathcal{M}(\mathbf{g})\mathbf{g}'}(\mathcal{E},\mathbf{k}_{\|})\Big)\Big\}O_{\mathbf{g}\mathbf{g}'}^{\mathbf{k}_{\|},V} \tag{6.101}$$

With these outputs of many different density of states from the code, we have an extensive tool to analyse our results and compare with lots of different spectroscopies.

6.3.2 Charge densities are hiding there too

In chapter 1 we already made a link between the Green function and the charge density. As we have already seen in (1.24), the following relation holds:

$$\rho(\mathbf{r},\mathbf{k}_{\|}) = \frac{1}{\pi}\mathrm{Im}\int_{-\infty}^{E_F}\mathrm{d}\mathcal{E}G(\mathbf{r},\mathbf{r};\mathcal{E},\mathbf{k}_{\|}) \tag{6.102}$$

This integral is usually performed along a semi-circle in the complex energy plane, starting from an energy low enough to include all valence states. Along this semi-circle a number of energy points is chosen for which we calculate the Green function matrix. (6.102) then becomes a sum:

$$\rho(\mathbf{r},\mathbf{k}_{\|}) = \frac{1}{\pi}\sum_i \mathrm{Im}\Big(w_i^{\mathcal{E}}G(\mathbf{r},\mathbf{r};\mathcal{E}_i,\mathbf{k}_{\|})\Big) \tag{6.103}$$

where $w_i^{\mathcal{E}}$ are the weights chosen so as to approximate the integral as closely as possible. The energies along the semi-circle and the weights are chosen according to Gaussian integration. For energy $\mathcal{E}$ we can write the following expression:

$$\rho(\mathbf{r};\mathcal{E}_i,\mathbf{k}_\parallel) = \frac{1}{\pi}\mathrm{Im}G(\mathbf{r},\mathbf{r};\mathcal{E}_i,\mathbf{k}_\parallel) \tag{6.104}$$

Now we can write $\rho(\mathbf{r},\mathcal{E}_i)$ throughout a sub-volume as:

$$\rho(\mathbf{r};\mathcal{E}_i,\mathbf{k}_\parallel) = \sum_{\mathbf{g}} \rho_{\mathbf{g}}(\mathcal{E}_i,\mathbf{k}_\parallel)e^{i\mathbf{g}\cdot\mathbf{r}} + \sum_{\alpha}\sum_{L''}^{n_l} \rho_{L''\alpha}(r;\mathcal{E}_i,\mathbf{k}_\parallel)Y_{L''}(\hat{\mathbf{r}})\theta(\mathbf{r}-\mathbf{r}_\alpha) \tag{6.105}$$

Let us now write out the expansion of $G(\mathbf{r},\mathbf{r};\mathcal{E}_i,\mathbf{k}_\parallel)$:

$$\begin{aligned}
G(\mathbf{r},\mathbf{r}) &= \sum_{\mathbf{g}\mathbf{g}'} \phi_{\mathbf{g}}^{\mathbf{k}_\parallel}(\mathbf{r})G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_\parallel)\phi_{\mathbf{g}'}^{\mathbf{k}_\parallel\,*}(\mathbf{r}) \\
&= \sum_{\mathbf{g}\mathbf{g}'}\left\{e^{i(\mathbf{g}+\mathbf{k}_\parallel)\cdot\mathbf{r}} + \sum_{\alpha}\sum_{L}^{n_l} \mathrm{P}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} f_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(r)Y_L(\hat{\mathbf{r}})\theta(\mathbf{r}-\mathbf{r}_\alpha)\right\}G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_\parallel)\times \\
&\qquad \left\{e^{-i(\mathbf{g}'+\mathbf{k}_\parallel)\cdot\mathbf{r}} + \sum_{\alpha'}\sum_{L'}^{n_l} \mathrm{P}_{L',\alpha'}^{\mathbf{g}',\mathbf{k}_\parallel\,*} f_{L',\alpha'}^{\mathbf{g}',\mathbf{k}_\parallel\,*}(r)Y_{L'}^*(\hat{\mathbf{r}})\theta(\mathbf{r}-\mathbf{r}_{\alpha'})\right\} \\
&= \sum_{\mathbf{g}\mathbf{g}'} G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_\parallel)\left\{e^{i(\mathbf{g}+\mathbf{k}_\parallel)\cdot\mathbf{r}}\right\}\left\{e^{-i(\mathbf{g}'+\mathbf{k}_\parallel)\cdot\mathbf{r}}\right\} + \\
&\quad \sum_{\mathbf{g}\mathbf{g}'} G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_\parallel)\left\{e^{i(\mathbf{g}+\mathbf{k}_\parallel)\cdot\mathbf{r}}\right\}\times \\
&\qquad \left\{\sum_{\alpha'}\sum_{L'}^{n_l} \mathrm{P}_{L',\alpha'}^{\mathbf{g}',\mathbf{k}_\parallel\,*} f_{L',\alpha'}^{\mathbf{g}',\mathbf{k}_\parallel\,*}(r)Y_{L'}^*(\hat{\mathbf{r}})\theta(\mathbf{r}-\mathbf{r}_{\alpha'})\right\} + \\
&\quad \sum_{\mathbf{g}\mathbf{g}'} G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_\parallel)\left\{\sum_{\alpha}\sum_{L}^{n_l} \mathrm{P}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} f_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(r)Y_L(\hat{\mathbf{r}})\theta(\mathbf{r}-\mathbf{r}_\alpha)\right\}\times \\
&\qquad \left\{e^{-i(\mathbf{g}'+\mathbf{k}_\parallel)\cdot\mathbf{r}}\right\} + \\
&\quad \sum_{\mathbf{g}\mathbf{g}'} G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_\parallel)\left\{\sum_{\alpha}\sum_{L}^{n_l} \mathrm{P}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel} f_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}(r)Y_L(\hat{\mathbf{r}})\theta(\mathbf{r}-\mathbf{r}_\alpha)\right\}\times \\
&\qquad \left\{\sum_{\alpha'}\sum_{L'}^{n_l} \mathrm{P}_{L',\alpha'}^{\mathbf{g}',\mathbf{k}_\parallel\,*} f_{L',\alpha'}^{\mathbf{g}',\mathbf{k}_\parallel\,*}(r)Y_{L'}^*(\hat{\mathbf{r}})\theta(\mathbf{r}-\mathbf{r}_{\alpha'})\right\}
\end{aligned} \tag{6.106}$$

The terms on the right-hand side of (6.106) we shall denote by (I), (II), (III), and (IV). Term (I) is the plane-wave density of states, and equating it to the first term in (6.105):

$$\sum_{\mathbf{g}''} \rho_{\mathbf{g}''}(\mathcal{E}_i,\mathbf{k}_\parallel)e^{i\mathbf{g}''\,\mathbf{r}} = \sum_{\mathbf{g}\mathbf{g}'} G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_\parallel)e^{i(\mathbf{g}-\mathbf{g}')\cdot\mathbf{r}} \tag{6.107}$$

we obtain:

$$\rho_{\mathbf{g}''}(\mathcal{E}_i,\mathbf{k}_\parallel) = \sum_{\mathbf{g}\mathbf{g}'} G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_\parallel)\delta_{\mathbf{g}'',(\mathbf{g}-\mathbf{g}')} \tag{6.108}$$

The angular part of the charge density expansion, the second term in (6 105), involves much more work

Let us first examine term (IV) in (6 106) We substitute the expressions for the augmentation functions $f^{\mathbf{g},\mathbf{k}_\parallel}_{L,\alpha}(r)$ and $f^{\mathbf{g}'\,\mathbf{k}_\parallel}{}^*_{L',\alpha'}(r)$ to obtain

$$\begin{aligned}(\mathrm{IV}) = \sum_{\mathbf{g}\mathbf{g}'} G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_\parallel) \sum_{\alpha}^{n_l}\sum_{L}\sum_{\alpha'}^{n_l}\sum_{L'} \mathrm{P}^{\mathbf{g},\mathbf{k}_\parallel}_{L,\alpha}\mathrm{P}^{\mathbf{g}',\mathbf{k}_\parallel}{}^*_{L',\alpha'} \times \\ \Big[A^{\mathbf{g},\mathbf{k}_\parallel}_{l,\alpha}\, A^{\mathbf{g}',\mathbf{k}_\parallel}{}^*_{l',\alpha'} u_{l,\alpha}(r)\, u_{l',\alpha'}{}^*(r) + A^{\mathbf{g},\mathbf{k}_\parallel}_{l,\alpha}\, B^{\mathbf{g}'\,\mathbf{k}_\parallel}{}^*_{l',\alpha'} u_{l,\alpha}(r)\, u^*_{l',\alpha'}(r) \\ - A^{\mathbf{g},\mathbf{k}_\parallel}_{l,\alpha} u_{l,\alpha}(r)\, \jmath_{l'}(|\mathbf{g}'+\mathbf{k}_\parallel|r) + B^{\mathbf{g},\mathbf{k}_\parallel}_{l,\alpha}\, A^{\mathbf{g}',\mathbf{k}_\parallel}{}^*_{l',\alpha'} u_{l,\alpha}(r)\, u_{l',\alpha'}{}^*(r) \\ + B^{\mathbf{g},\mathbf{k}_\parallel}_{l,\alpha}\, B^{\mathbf{g}',\mathbf{k}_\parallel}{}^*_{l',\alpha'} u_{l,\alpha}(r)\, u^*_{l',\alpha'}(r) - B^{\mathbf{g}\,\mathbf{k}_\parallel}_{l,\alpha}\, u_{l\,\alpha}(r)\, \jmath_{l'}(|\mathbf{g}'+\mathbf{k}_\parallel|r) \\ - A^{\mathbf{g}',\mathbf{k}_\parallel}{}^*_{l',\alpha'} \jmath_l(|\mathbf{g}+\mathbf{k}_\parallel|r)\, u_{l',\alpha'}{}^*(r) - B^{\mathbf{g}',\mathbf{k}_\parallel}{}^*_{l'\,\alpha'} \jmath_l(|\mathbf{g}+\mathbf{k}_\parallel|r)\, u^*_{l',\alpha'}(r) \\ + \jmath_l(|\mathbf{g}+\mathbf{k}_\parallel|r)\, \jmath_{l'}(|\mathbf{g}'+\mathbf{k}_\parallel|r)\Big] \times Y_L(\hat{\mathbf{r}})Y^*_{L'}(\hat{\mathbf{r}})\theta(\mathbf{r}-\mathbf{r}_\alpha)\theta(\mathbf{r}-\mathbf{r}_{\alpha'})\end{aligned} \tag{6 109}$$

The fact that this looks rather complicated is a direct result of the use of the new LAPW With the old LAPW, we would only have the terms in (6 109) with two A's or B's and two atomic solutions Now we have 5 extra terms That these 5 terms are more than just a bit of extra work will be discussed later

To examine terms (II) and (III) in (6 106), we expand the plane waves into spherical harmonics around the centre of atom α using (6 12) Term (II) then becomes

$$\begin{aligned}(\mathrm{II}) = \sum_{\mathbf{g}\mathbf{g}'} G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_\parallel) \sum_{\alpha}^{\infty}\sum_{L}\sum_{\alpha'}^{n_l}\sum_{L'} \mathrm{P}^{\mathbf{g},\mathbf{k}_\parallel}_{L,\alpha}\mathrm{P}^{\mathbf{g}',\mathbf{k}_\parallel}{}^*_{L',\alpha'} \times \\ \Big[A^{\mathbf{g}',\mathbf{k}_\parallel}{}^*_{l',\alpha'} \jmath_l(|\mathbf{g}+\mathbf{k}_\parallel|r)\, u_{l',\alpha'}{}^*(r) + B^{\mathbf{g}',\mathbf{k}_\parallel}{}^*_{l',\alpha'} \jmath_l(|\mathbf{g}+\mathbf{k}_\parallel|r)\, u^*_{l',\alpha'}(r) \\ - \jmath_l(|\mathbf{g}+\mathbf{k}_\parallel|r)\, \jmath_{l'}(|\mathbf{g}'+\mathbf{k}_\parallel|r)\Big] \times Y_L(\hat{\mathbf{r}})Y^*_{L'}(\hat{\mathbf{r}})\theta(\mathbf{r}-\mathbf{r}_{\alpha'})\end{aligned} \tag{6 110}$$

Similarly term (III) becomes

$$\begin{aligned}(\mathrm{III}) = \sum_{\mathbf{g}\mathbf{g}'} G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_\parallel) \sum_{\alpha}^{n_l}\sum_{L}\sum_{\alpha'}^{\infty}\sum_{L'} \mathrm{P}^{\mathbf{g},\mathbf{k}_\parallel}_{L,\alpha}\mathrm{P}^{\mathbf{g}',\mathbf{k}_\parallel}{}^*_{L',\alpha'} \times \\ \Big[A^{\mathbf{g},\mathbf{k}_\parallel}_{l,\alpha}\, u_{l,\alpha}(r)\, \jmath_{l'}(|\mathbf{g}'+\mathbf{k}_\parallel|r) + B^{\mathbf{g},\mathbf{k}_\parallel}_{l,\alpha}\, u_{l,\alpha}(r)\, \jmath_{l'}(|\mathbf{g}'+\mathbf{k}_\parallel|r) \\ - \jmath_l(|\mathbf{g}+\mathbf{k}_\parallel|r)\, \jmath_{l'}(|\mathbf{g}'+\mathbf{k}_\parallel|r)\Big] \times Y_L(\hat{\mathbf{r}})Y^*_{L'}(\hat{\mathbf{r}})\theta(\mathbf{r}-\mathbf{r}_\alpha)\end{aligned} \tag{6 111}$$

Now we notice that there is some cancellation of terms

- the first n_l l's of the first term in (6 110) cancel with the 7^{th} term in (6 109)

- the first n_l l's of the second term in (6.110) cancel with the 8^{th} term in (6.109)
- the first n_l l's of the third term in (6.110) cancel with the 9^{th} term in (6.109)
- the first n_l l''s of the first term in (6.111) cancel with the third term in (6.109)
- the first n_l l''s of the second term in (6.111) cancel with the 6^{th} term in (6.109)
- the first n_l l''s of the third term in (6.111) returns the 9^{th} term of (6.109) to the expression with opposite sign.

From the expression for the charge density in (6.105) we obtain:

$$\rho_{L''\alpha}(r,\mathcal{E}_i,\mathbf{k}_\parallel) = \int_\alpha \mathrm{d}^3\mathbf{r}\left[\rho(\mathbf{r};\mathcal{E}_i,\mathbf{k}_\parallel) - \sum_{\mathbf{g}} \rho_{\mathbf{g}}(\mathcal{E}_i,\mathbf{k}_\parallel)e^{i\mathbf{g}\cdot\mathbf{r}}\right] Y^*_{L''}(\hat{\mathbf{r}}) \tag{6.112}$$

So we only use the second term on the right-hand-side of (6.105). When we apply (6.112) to the terms that we get from (II), (III) and (IV), we obtain the following expression for the muffin-tin radial expansion of the charge-density:

$$\begin{aligned}
\rho_{L''\alpha}(r;\mathcal{E},\mathbf{k}_\parallel) = \sum_{\mathbf{g}\mathbf{g}'} G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_\parallel) \sum_\alpha \sum_L^{n_l} \sum_{L'}^{n_l} \mathrm{P}^{\mathbf{g},\mathbf{k}_\parallel}_{L,\alpha} \mathrm{P}^{\mathbf{g}',\mathbf{k}_\parallel\,*}_{L',\alpha} \times \\
\Big[A^{\mathbf{g},\mathbf{k}_\parallel}_{l,\alpha}\, A^{\mathbf{g}',\mathbf{k}_\parallel\,*}_{l',\alpha}\, u_{l,\alpha}(r)\, u_{l',\alpha}{}^*(r) \\
+ A^{\mathbf{g},\mathbf{k}_\parallel}_{l,\alpha}\, B^{\mathbf{g}',\mathbf{k}_\parallel\,*}_{l',\alpha}\, u_{l,\alpha}(r)\, \dot{u}^*_{l',\alpha}(r) \\
+ B^{\mathbf{g},\mathbf{k}_\parallel}_{l,\alpha}\, A^{\mathbf{g}',\mathbf{k}_\parallel\,*}_{l',\alpha}\, \dot{u}_{l,\alpha}(r)\, u_{l',\alpha}{}^*(r) \\
+ B^{\mathbf{g},\mathbf{k}_\parallel}_{l,\alpha}\, B^{\mathbf{g}',\mathbf{k}_\parallel\,*}_{l',\alpha}\, \dot{u}_{l,\alpha}(r)\, \dot{u}^*_{l',\alpha}(r) \\
- j_l(|\mathbf{g}+\mathbf{k}_\parallel|r)\, j_{l'}(|\mathbf{g}'+\mathbf{k}_\parallel|r)\Big] \times C^{L''}_{L,L'} \\
+ \sum_{\mathbf{g}\mathbf{g}'} G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_\parallel) \sum_\alpha \sum_{L=n_l+1}^{\infty} \sum_{L'}^{n_l} \mathrm{P}^{\mathbf{g},\mathbf{k}_\parallel}_{L,\alpha} \mathrm{P}^{\mathbf{g}',\mathbf{k}_\parallel}_{L',\alpha} \times \\
\Big[A^{\mathbf{g}',\mathbf{k}_\parallel\,*}_{l',\alpha'}\, j_l(|\mathbf{g}+\mathbf{k}_\parallel|r)\, u_{l',\alpha'}{}^*(r) \\
+ B^{\mathbf{g}',\mathbf{k}_\parallel\,*}_{l',\alpha'}\, j_l(|\mathbf{g}+\mathbf{k}_\parallel|r)\, \dot{u}^*_{l',\alpha'}(r) \\
- j_l(|\mathbf{g}+\mathbf{k}_\parallel|r)\, j_{l'}(|\mathbf{g}'+\mathbf{k}_\parallel|r)\Big] \times C^{L''}_{L,L'}
\end{aligned}$$

$$+\sum_{\mathbf{g}\mathbf{g}'} G_{\mathbf{g}\mathbf{g}'}(\mathcal{E},\mathbf{k}_\parallel)\sum_{\alpha}\sum_{L}^{n_l}\sum_{L'=n_l+1}^{\infty} \mathrm{P}_{L,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}\mathrm{P}_{L',\alpha'}^{\mathbf{g}',\mathbf{k}_\parallel *}\times$$
$$\Big[A_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}\, u_{l,\alpha}(r)\, j_{l'}(|\mathbf{g}'+\mathbf{k}_\parallel|r)$$
$$+ B_{l,\alpha}^{\mathbf{g},\mathbf{k}_\parallel}\, \dot{u}_{l,\alpha}(r)\, j_{l'}(|\mathbf{g}'+\mathbf{k}_\parallel|r)$$
$$- j_l(|\mathbf{g}+\mathbf{k}_\parallel|r)\, j_{l'}(|\mathbf{g}'+\mathbf{k}_\parallel|r)\Big]\times C_{L,L'}^{L''} \qquad (6.113)$$

Because of the triangle rule[13, 14] for the Clebsch-Gordon coefficients $C_{L,L'}^{L''}$, the infinite sums in (6.113) become finite sums.

6.3.2.1 The new LAPWs are VERY expensive

Upon closer examination of (6.113), we see some rather unpleasant terms. All the terms with the Bessel functions depend on both the wave-vector of the plane-wave $\mathbf{g}$, and on the radial position, r. Using the old LAPW basis there were none of these type of terms. The only terms which showed up there were terms that depended on the radial position, r, i.e. only terms with radial solutions. Since the radial solutions are the same for all $\mathbf{g}$, this meant that the radial expansion of the charge-density could be calculated rather quickly. In fact in the old implementation of embedding theory, using the traditional LAPW's, the subroutine which calculates the radial expansion of the charge-density, `MTRHO`, was one of the fastest.

However, now that we use the new LAPW's, most of the computing time in the `GROW` module is spent in subroutine `MTRHO` - if we implement (6.113) in a straightforward manner. This is because we now have several nested loops over wave-vectors and radial position. Using complicated tricks which I shall not discuss in this thesis, I have actually managed to get the calculation time spent in `MTRHO` down drastically, but `MTRHO` still remains one of the most time-consuming subroutines.

Once we have the muffin-tin radial expansion for each energy $\mathcal{E}_i$, we can add these expansions, weighted with $w_i^{\mathcal{E}}$, to get the radial expansion of the total charge-density. In fact, the total charge density can now be written as:

$$\begin{aligned}\rho(\mathbf{r}) &= \int_{\mathrm{BZ}} \mathrm{d}^2\mathbf{k}_\parallel \rho(\mathbf{r},\mathbf{k}_\parallel)\\ &= \int_{\mathrm{BZ}} \mathrm{d}^2\mathbf{k}_\parallel \frac{1}{\pi}\sum_i \mathrm{Im}\Big(w_i^{\mathcal{E}} G(\mathbf{r},\mathbf{r};\mathcal{E}_i,\mathbf{k}_\parallel)\Big)\end{aligned} \qquad (6.114)$$

We finally replace the integral over the Brillouin zone by a finite sum:

$$\rho(\mathbf{r}) = \frac{1}{\pi}\mathrm{Im}\sum_n\sum_i w_n^{\mathbf{k}_\parallel} w_i^{\mathcal{E}} G(\mathbf{r},\mathbf{r};\mathcal{E}_i,\mathbf{k}_{\parallel_n}) \qquad (6.115)$$

So we have managed to get the charge-density from the Green function. The charge-density is one of the major inputs to the module `FISH` which we shall discuss in the next chapter.

6.4 The structure of the GROW module

In figures 3.13 and 3.14 in chapter 3 I have already given an overview of the structure of the GROW module, although there it was framed in terms of a general application of embedding. In this section I shall indicate the structure of the GROW module in increasing detail.

The overall structure of the GROW module can be split into three parts, input & initialisation, calculation and output, although some input/output may be performed during the calculation part.

6.4.1 Input and initialisation

Let us first concentrate on the input and initialisation part. In this section I will use the real names of subroutines where possible. In figure 6.1 a flow-chart of the input/initialisation part is shown.

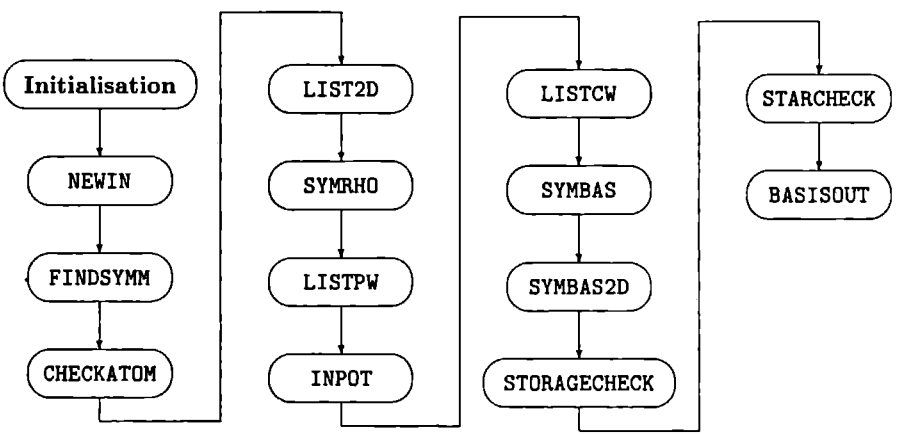


Figure 6.1: *Small scale structure of input and initialisation part of the* GROW *module.*

The meaning of the different modules in this figure is:

- **Initialisation**: Files are opened and variables are initialised. Some opening of files and initialisation of variables is done in the other subroutines or in between.
- NEWIN: The structure of the material we are interested in is input as well as most control parameters. Some of these control parameters are:
  - `itype`: controls whether calculation will be bulk(0), surface(1), or interface(2).
  - `basismode`: mode in which the entire code is run, using either embedded plane waves (a future basis set possibility, not used at present) (0), linear augmented plane waves (1) or dragged linear augmented plane waves (2). The last mode is

one where the pivot energies for the LAPW are taken to be the energy we are working at.

- `basistype`: controls the shape of the wave-vectors chosen for the LAPW's. When `basistype` equals 0, the wave-vectors are sorted according to length (i.e. spherical distribution), when 1 the wave-vectors are constructed for different parallel vectors with a given number of perpendicular components (i.e. cylindrical distribution). When `basistype` equals 2 the distribution is read in from a file.
- `relat`: when 1 all semi-relativistic corrections are switched on, otherwise switched off.
- `ldos`: when 1 angular momentum l-resolved DOS is output.
- `orbdos`: when 1 angular momentum (l, m)-resolved DOS is output.
- `mirror`: when 1 mirror-symmetry-resolved DOS is output. In this case some extra subroutines have to be inserted in the flow of figure 6.1, but this is a detail we shall not discuss.

- FINDSYMM: This routine checks the set of symmetry operations which build the point-group of the lattice and the point-group of the lattice + atomic basis.
- CHECKATOM: Distances between all atoms are checked against muffin-tin radii, to ensure that no muffin-tins intersect.
- LIST2D: The planar wave-vectors are constructed in stars. We always use complete stars, contrary to the old SEGF code.
- SYMRHO: This routine finds the angular momentum components which are allowed by symmetry to be non-zero. It also sets up some arrays to be used later to symmetrize the charge-density.
- LISTPW: Constructs the entire list of plane waves, using output from LIST2D and the `basistype` variable.
- INPOT: Inputs the potential from the file `"pot"`. Contained in this file is the potential expansion and the plane wave basis in which the interstitial potential is expanded.
- LISTCW: Constructs the plane wave basis used to expand the charge-density, by combining wave-vectors of the LAPW's, as indicated in (6.107) and (6.108).
- SYMBAS: Takes the charge-density wave-vectors and constructs the three-dimensional stars of symmetry equivalent vectors.
- SYMBAS2D: Similar to SYMBAS, but now for two-dimensional stars.
- STORAGECHECK: Checks whether enough temporary storage has been defined. Also performs the indexing trick for the calculation of the matrix elements of the interstitial potential (see page 89).
- STARCHECK: This routine checks whether the two-dimensional stars constructed in SYMBAS2D are the same in all subvolumes. They must be, to be able to use embedding techniques to couple subvolumes.

- BASISOUT: This routine outputs the LAPW and charge-density basis.

Once this part of the code is finished, everything has been set up for the possible tracks the code can take from here. Everything that has been mentioned in this section will be done for whatever mode the code is run in. In the next section we will go into the details of the calculation part.

6.4.2 Structure of the calculation in GROW

The structure of the GROW module becomes a lot more complicated once we have passed the input/initialisation part. Since this same code can be used to calculate several different things, it is run using several modes. One of the parameters to control this mode is input in NEWIN, but has not been mentioned in the previous section. The control parameter is called **emode**. Let us now examine its possible values:

- **emode** = -1: this means that we want a band-structure, for one single $\mathbf{k}_{\|}$. This also means that the energy-grid (where the parameter originally gets its name) is a line along the real axis, at a particular imaginary part of the energy, which is read in.

- **emode** = 0: this is the value requesting a density of states calculation (if only one $\mathbf{k}_{\|}$ is specified a band-structure is also calculated). The energy grid is the same as for **emode** equal to minus one.

- **emode** = 1: a charge-density calculation is performed, with fixed (and read-in) Fermi energy. The energy grid is a semi-circle contour from the starting energy (input) to the Fermi energy.

- **emode** = 2: similar to **emode** = 1, but now an iterative process is started to vary the Fermi energy until the system is charge neutral. This is the mode most often used in self-consistent bulk calculations. The energy grid is adaptive and changes dynamically with the Fermi energy, i.e. a new distribution on a semi-circle in the complex energy plane is calculated for each approximation to the Fermi energy. This we will call the fermi-search option.

- **emode** = 3: a single energy (complex) and $\mathbf{k}_{\|}$ are input and the $\mathbf{k}_{\|}$-resolved local density of states (charge-density at single energy) is calculated. There is no energy grid.

- **emode** = 4: only the embedding potential is calculated on a semi-circle contour in the complex energy plane.

- **emode** = 5: only the embedding potential is calculated but now on a line parallel to the real axis in the complex energy plane.

I shall give an overview of the structure of the calculation for the first five of these different modes. The last two that only calculate the embedding potentials are essentially special cases of the others.

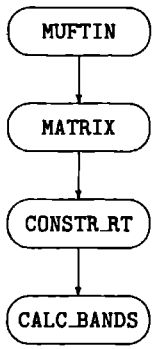


Figure 6.2: *Large scale structure within subroutine* DOBAND.

6.4.2.1 The band-structure calculation

In the case of a band-structure calculation a call is made to the subroutine BANDS. This routine does some initialisation and then for each energy along the line in the complex energy plane calls the routine DOBAND. The main structure of the band-structure calculation is in this routine DOBAND. In figure 6.2 I give the overall structure within DOBAND. Let us discuss the routines shown in figure 6.2 in more detail:

- The routine MUFTIN solves the Dirac equation in all muffin-tins. This is done by calls to ATOM_LAPW. Output of the MUFTIN routine are all the atomic solutions and energy-derivatives $u_l(r)$ and $\dot{u}_l(r)$ and all of the integrals discussed in appendix B containing these atomic solutions. This routine is called for every energy, but only performs actual calculations for the first energy.

- The routine MATRIX constructs the Hamiltonian and overlap matrices in all subvolumes. All these matrix elements are discussed in section 6.2. MATRIX performs calls to different other subroutines. Some of these other subroutines are:

  - INTWRP: to calculate the interstitial warping potential matrix. See section 6.2.3.
  - FINDNDGG: this routine does some initialisation of arrays needed in the speed-up tricks used in other routines.
  - MTME: returns the Hamiltonian and overlap matrix elements in the muffin-tin (for the spherically symmetric part of the potential). See section 6.2.1.
  - MTWP: returns the matrix elements of the muffin-tin warping potential. See section 6.2.2.
  - INTME: returns the interstitial kinetic energy and overlap matrix elements. This routine performs the shift of the embedding surface to the embedding plane as discussed in section 2.5.

  The routine MATRIX is called for every energy, but like the routine MUFTIN, only performs actual calculations for the first energy, because all the matrix elements are energy-independent.

- The routine CONSTR_RT performs most of the calculations discussed in sections 3.4.1, 3.4.2.1, 3.4.2.2, and 3.4.2.3. First a call is made to GETRT. This subroutine does the following:

  - First a call to GETG to obtain the Green function with zero normal derivative on both embedding planes in each subvolume. See section 6.3.
  - This Green function is used in each subvolume to calculate the scattering matrices: R_L, R_R, T_L, and T_R. This is described in section 3.4.1.

  The routine CONSTR_RT then shifts the origin of the scattering matrices as described in section 3.4.2.1. The shifted origin scattering matrices are then used to construct the transfer matrix for each subvolume as described in section 3.4.2.2. Finally the transfer matrices of the separate subvolumes are combined to get the transfer matrices (left-to-right and right-to-left) of the entire stack of subvolumes.

- The subroutine CALC_BANDS uses the transfer matrix of the stack of subvolumes which form the bulk-repeat-unit, to find the band-structure. As already mentioned in section 4.2 the band-structure is the set of eigenvalues of the transfer matrix. In general we are only interested in the real bands, thus for each energy the k_z values with imaginary part less than a certain tolerance are written out in the band-structure output-file.

As we have seen, the band-structure calculation doesn't actually make use of embedding theory directly. Very extensive use is made of the scattering theory framework constructed in chapter 3.

6.4.2.2 The density of states calculation

The structure of the density of states calculation is rather similar to the band-structure calculation. The energies also run along a line parallel to the real axes of the complex energy plane. The routine that controls the DOS calculation is DOITER. This routine contains both the $\mathbf{k}_{\|}$ loop and the energy loop. For each $\mathbf{k}_{\|}$ and energy the routine DOENERGY is called. The large scale structure of DOENERGY is shown in figure 6.3. The first two routines called we have already seen in detail in the previous section. Again these are called for all energies, but action is taken only for the first energy. In this case, however, the routine MATRIX has to perform the calculation for every $\mathbf{k}_{\|}$, because the matrix-elements depend on this vector. Because the $\mathbf{k}_{\|}$-loop is the outer loop in DOITER, this means that each time we call MATRIX with the first energy of the energy loop, we have to recalculate the matrix-elements.

The routine GETG we have seen before as well. This time the routine is called with the mode which will give the Green function with an embedding potential on both embedding planes. I shall now discuss the routines we have not seen before.

- The routine CONSTR_EMBPOT is the routine which will eventually give us the embedding potential on both sides of each subvolume. The action of CONSTR_EMBPOT is controlled by several control parameters. One of them is the itype parameter mentioned before in section 6.4.1. The other three are:

  - iemb: This parameter controls the type of embedding potential we want on each side of the subvolume. The different possibilities are:

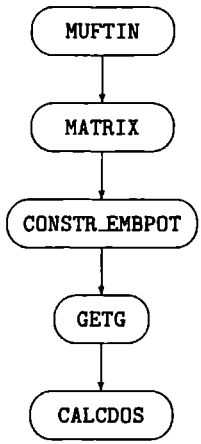


Figure 6.3: *Large scale structure withing subroutine* DOENERGY.

  - * iemb = 0 or 1: find the embedding potentials to the right and left side of the bulk-repeat-unit by the process of iteration, as discussed in sections 3.4.2.4, 3.4.2.5, and 3.4.2.6.
  - * iemb = 2: the embedding potentials to the right and left of the stack of subvolumes are read in from file.
  - * iemb = 3: the embedding potential on the right of the stack is read in from file, the embedding potential on the left is calculated using the analytic expression for the vacuum embedding potential.
  - * iemb = 4: similar to 3, but now the vacuum is on the right and the left embedding potential is read in from file.
  - * iemb = 5: both embedding potentials are calculated using the analytic expression for the vacuum embedding potential.
- iread: When reading or iterating the embedding potentials, this parameter controls the actions taken. The possible values are:
  - * iread = 0: iterate and start with zero matrix
  - * iread = 1: read in from file
  - * iread = 2: read in for first energy, then use the result for one energy as starting point for the iteration for the next energy.
  - * iread = 3: use zero matrix as iteration starting point for the first energy, for the other energies use the result of the previous energy as starting point for the iteration of the embedding potentials.
- iwrite: This parameter controls the output of embedding potentials for use in other calculations. Possible values are:
  - * iwrite = 0: no embedding potentials are output.
  - * iwrite = 1: only the embedding potential to the left of the entire stack is output.

* `iwrite = 2`: only the embedding potential to the right of the entire stack is output.
* `iwrite = 3`: combination of 1 & 2, both embedding potentials are output.
* `iwrite = 4`: same as 3, but now also output of sclected embedding potentials of chosen subvolumes.

In all cases not only the embedding potential is output, but also the reflection matrix from which the embedding potential is derived as described in section 3.4.2.6.

Depending on these parameters, the routine `CONSTR_EMBPOT` will use the techniques described in section 3.4.2 to obtain embedding potentials to the right and left of each subvolume. To do this the routine will call other subroutines. Some of these other subroutines are:

- `CONSTR_RT`: as discussed in the previous section.
- `GETRVAC`: calculates the vacuum embedding potential for the given energy and $\mathbf{k}_{||}$.
- `BULKREP`: performs the iterative process by stacking the bulk-repeat-unit, using the procedures outlined in sections 3.4.2.4, 3.4.2.5, and 3.4.2.6.
- `PROPAGATE`: propagates the embedding potentials on the sides of the stack of subvolumes to each separate subvolume, using the method described in section 3.4.2.7.

- The subroutine `CALCDOS` is fairly simple. It just performs the summations described in (6.91) for various types of overlap matrix. Then it also adds the DOS contribution of a given $\mathbf{k}_{||}$ to the total DOS.

At the end of both the $\mathbf{k}_{||}$ and energy loops the program will have output the different densities of states we have requested.

6.4.2.3 The charge-density calculation

For the charge-density calculation, a different energy grid is needed. This grid is constructed by the routine `EGRID`. The energies selected lie on a semi-circle starting at a given energy (below the bottom of the valence band) and ending at the given Fermi energy.

Before we can calculate the valence charge-density as defined in (6.115), we will have to calculate the core charge-density. The core states are calculated in the routine `CORDEN`. This routine performs a loop over all the core-states and for each calls the routine `CORE`. The routine `CORE` solves either the semi-relativistic Schrödinger equation or the Dirac equation. In the case of the semi-relativistic equation, the boundary for the core-states on the muffin-tin is zero amplitude (i.e. the core states do not 'leak' out of the muffin-tin), and they are normalised to one inside the muffin-tin. When we solve the Dirac equation for the core-states, the solutions are extended outside the muffin-tin, assuming a constant potential, and the extended solution is normalised. In that case a part of the core-state may lie outside the muffin-tin (The difference in treatment of the semi-relativistic and Dirac equations is historic). The charge-density of the core states is added to the radial charge-density expansion, so that the total charge-density should give the atomic number per atom.

The calculation of the valence charge-density is also performed by the routine `DOITER`, using the same loop structure as for the DOS calculation. The structure of the charge-density calculation is the same as shown in figure 6.3, with the routines `MTRHO` and `INTRHO` added after the routine `CALCDOS`. Let us examine these last two subroutines in a bit more detail.

- The subroutine INTRHO performs the calculation of the interstitial charge-density from the Green function, as described in (6.107) and (6.108).

- In routine MTRHO the muffin-tin charge-density is calculated, using the expression given by (6.113). Some tricks are applied in this routine to speed up the calculation. I shall not go into the details here, but will only mention that the tricks involve calculation of the charge-density on a smaller grid and then interpolating to the larger grid used in the rest of the code, and the use of a non-negligible amount of storage, to prevent calculating quantities more than once.

Once the $\mathbf{k}_{\|}$ and energy loops are performed we have the charge-density expansion. This charge-density expansion is then symmetrized in the muffin-tin by BZMT and in the interstitial by BZG.

6.4.2.4 The charge-density calculation with Fermi energy search

In the previous section we have calculated the charge-density expansion for a fixed Fermi energy. However, we do not always know the correct Fermi energy in advance. Especially when we are calculating the bulk potential self-consistently, the Fermi energy will change with the changing potential. To enable the GROW code to find the Fermi energy, a routine has been written around the previously mentioned DOITER, called FERMI_SEARCH. As the name implies, this is the routine which searches for the correct Fermi energy, i.e. the Fermi energy for which the total charge of the entire subvolume stack is zero. The total charge in each subvolume is given by:

$$q_i = \sum_{\alpha} \left(q^{\alpha}_{\text{core}} + q^{\alpha}_{\text{val}} - Z_{\alpha} \right) + q^{\text{int}}_{\text{val}} \tag{6.116}$$

where i indicates the subvolume, q^{α}_{core} is the core-charge associated with muffin-tin α (this is an integer number, even when there is core-state 'leakage'), q^{α}_{val} is the valence charge in muffin-tin α, Z_{α} is the atomic number of muffin-tin α, and $q^{\text{int}}_{\text{val}}$ is the interstitial valence charge. The total charge q_{tot} is given by:

$$q_{\text{tot}} = \sum_{i} q_i \tag{6.117}$$

The search for the Fermi energy goes as follows:

1. Make a guess for the Fermi energy.

2. Create an energy grid using EGRID.

3. Call DOITER and calculate q_{tot}.

4. If q_{tot} is zero, we have completed the Fermi energy search.

5. Otherwise, if q_{tot} is positive, our trial Fermi energy is too high and we have to try a lower one. If q_{tot} is negative, our trial Fermi energy is too low and we have to try a higher one.

6. Once we have bounded the Fermi energy, i.e. we have two trial values, for one of which the total charge is positive and for the other negative, we can use elaborate schemes to find a new trial Fermi energy. Before we have found the bounds we will increase or decrease the Fermi energy by a given amount, each time doubling this amount as long as no bounds have been found. Once we have a new trial Fermi energy, we jump back to step 2.

It is clear that this can be an expensive process, because for each trial value of the Fermi energy an entire $\mathbf{k}_{\|}$ and energy loop has to be executed. Even though we do not need to recalculate the matrix elements (as we shall see in the next paragraph), and we do not need to calculate the charge-density (see later on) this still takes time because we have to invert matrices. Therefore once we have bounds for the Fermi energy, we want a scheme for finding the trial Fermi energies which is efficient. The algorithm that has been implemented is Brent's method[16, 17].

Another potential source of increased computing time, is that for each try, the Hamiltonian matrix elements for each $\mathbf{k}_{\|}$ would be calculated again, although these do not change with changing Fermi energy. The trick I have implemented here is that the Hamiltonian and overlap matrices are written to a file for each $\mathbf{k}_{\|}$, and are read in the next time they would normally be re-calculated. This means that only the first trial Fermi energy is expensive in that part of the calculation.

The most important improvement in computing time for the Fermi energy search, however, is the fact that the total charge can be calculated from other sources than the total charge-density expansion. This means that MTRHO only needs to be called for the final correct Fermi energy.

With all these speed-up tricks and the efficient search algorithm, the search for the correct Fermi energy and the calculation of the total charge-density expansion for the Fermi energy has become quite efficient.

6.4.2.5 The local density of states calculation

This particular calculation of the local density of states involves nothing new. Since the local density of states is equal to the charge-density at a particular energy and $\mathbf{k}_{\|}$, and since we already perform that calculation withing the $\mathbf{k}_{\|}$ and energy loops in DOITER, we only need to call DOITER with both loops set to length one. An important difference is that for the case of the single energy and single $\mathbf{k}_{\|}$ the charge-density (or local density of states) should not be symmetrized. Therefore there are no calls to BZMT and BZG.

Now that we have discussed how all the physical quantities we might be interested in are to be calculated, we shall have to pass them on to the outside world.

6.4.3 Output

Output of both physical quantities and results needed for other calculations is performed in several routines. I name some of them here, together with their function.

- CALC_BANDS: outputs the band-structure.
- CONSTR_EMBPOT: outputs the embedding potentials and reflections matrices.

- DOSOS outputs all sorts of densities of states
- GETMULTIP calculates and outputs the charge-density multipoles, to be used and defined in chapter 7
- RHOOUT outputs the charge-density expansion

In this section I have outlined the structure of the code In doing this I have mentioned what the code can calculate, what input is needed, and what is output In the next section I will describe the input and output in more detail

6.5 Raw materials going into the GROW module

In this section the input files are described in some detail, though not fully - this is not a manual

6.5.1 The "struct" file

In the struct file, the structure and geometry is given for the material which we are calculating, together with several control parameters This file acts as input for all the modules, i e for the GROW module, as well as for the FISH and MIX modules
Some of the variables describing the structure are

- basis vectors of the parallel lattice, shared by all subvolumes,
- number of subvolumes in the stack,
- number of atoms and positions of atoms in each subvolume,
- muffin-tin radii of atoms,
- vectors linking the origins in the different subvolumes

Other parameters in the file are

- sort of calculation to perform, i e bulk, surface, or interface,
- type of basis to use,
- what types of DOS output we want,
- several tolerances used throughout the code,
- number of basis functions to use (LAPW's),
- number of angular momenta to be used in augmentation,
- number of angular momenta to be used in charge-density expansion

6.5.2 The "ekgrid" file

The ekgrid file contains further variables as well as control parameters. Amongst others it contains some previously mentioned: `emode`, `iemb`, `iread`, and `iwrite`. Other variables input in ekgrid are:

- number of $\mathbf{k}_{\|}$ points required,
- number of energies in the energy grid,
- starting energy of the energy grid,
- final energy of the energy grid, the Fermi energy,
- imaginary energy to use in the case of a straight line energy grid,
- increment for increasing Fermi energy, before bounds are found in Fermi energy search,
- several tolerances.

6.5.3 The "pot" file

This file contains the potential expansion information in each subvolume as well as some other variables. Some of these are:

- atomic number for each atom,
- size of grid for each atom,
- grid parameter d_x for each atom (see (B.5) in appendix B),
- number of core states for each atom,
- angular momentum of all core states and estimates for their energies,
- pivot energies E_l for each atom.

6.5.4 The "rmat*" and "embpot*" files

These files contain embedding potential matrices or reflection matrices. For each of the matrices in the file the size of the matrix, as well as the exact energy and $\mathbf{k}_{\|}$ for which the matrix was originally calculated is stored as well.

I have mentioned some of the raw materials going into the GROW code. Using these raw materials the code is able to grow products, some of which are physical quantities we are interested in, and some are input for other calculations.

6.6 Finished products grown by the GROW module

In this section I shall indicate the different types of finished products in the different output files. For the sake of completeness there is some repetition of section 6.5.4.

6.6.1 The "bands.*" files

The band-structure is contained in these files. The bands.all file contains the total complex band-structure. The real band-structure is stored in bands.real. In the case of a mirror-symmetry at a surface, the even real bands are stored in bands.even, whereas the odd real bands are stored in bands.odd.

6.6.2 The "dos.sv*" files

All the types of density of states which the code can grow are written into different files, one file for each subvolume, indicated by sv? in the name, with ? the number of the subvolume. Also the ending after this subvolume number indicates the type of DOS:

- .emb: DOS in embedded region
- .tot: DOS in total region. Difference with previous file is that the protruding caps from neighbouring subvolumes are included
- .mt: DOS in muffin-tins
- .l: angular momentum l-resolved DOS in each muffin-tin
- .orb: angular momentum (l, m)-resolved DOS in each muffin-tin
- .eo: even/odd resolved DOS in embedded region
- .mteo: even/odd resolved DOS in each muffin-tin

So, for example, the file dos.sv3.orb contains the (l, m)-resolved DOS in each muffin-tin in subvolume 3.

6.6.3 The "rho.*" and "qlm.sv*" files

These files contain the information about the charge-density. The total muffin-tin charge-density expansion in each muffin-tin is stored in the file rho.mt. This file contains the expansion in all muffin-tins for all subvolumes in the stack. The charge-density expansion in the interstitial in each subvolume of the stack is stored in rho.int. The multipole moments in the muffin-tins of subvolume n are stored in the file qlm.sv(n).

In this chapter I have given a description of the implementation of the embedding framework in the GROW module. This is the most important module, because it contains all the new embedding and scattering ideas. It is also important because it is the module which gives us the quantities we want to compare to experiment. In the next chapter we shall look at the implementation of the solution of Poisson's equation.

Chapter 7

FISHing using Poisson's equation

In this chapter I shall discuss the second module of the embedding code, the FISH module. This module solves Poisson's equation. We shall see what this means in our layered geometry. As already mentioned for the GROW module, it is impossible to go into all the details, simply because of the size of the module in terms of computer code. The FISH module is about 8500 lines of FORTRAN. At the end of this chapter we shall look at the structure of the FISH module and its input and output. First let us look at Poisson's equation, and define the problem in terms of our description of the physical system.

7.1 Poisson's equation

Poisson's equation relates the potential to the charge-density:

$$\nabla^2 V(\mathbf{r}) = -4\pi\rho(\mathbf{r}) \tag{7.1}$$

This is the equation which the FISH module solves for the layered geometry.

Solving this equation for a periodic charge-density is in principle rather straightforward. Normally, the idea is to first look at the short-range Coulomb interactions coming from neutral charge distributions and then use lattice-sums to add the long-range interactions. The lattice sums were first considered by Madelung[18] (the potential arising from the lattice summation is sometimes called the Madelung potential), who obtained the electrostatic potential and energy of a lattice. A more general method for obtaining the lattice sums was developed by Ewald[19] (the method for obtaining the potential from the lattice sum is sometimes called Ewald summation).

The method I shall use here is the method developed by Weinert[10], using the concept of multipole potentials and the Dirichlet boundary-problem for a sphere.

7.2 Solution of Poisson's equation

First, consider a distribution of charge, $\rho(\mathbf{r})$, localised inside a sphere S. We can then write the potential outside the sphere in terms of the multipole expansion[20]:

$$V(\mathbf{r}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{4\pi}{2l+1} Q_{lm} \frac{Y_{lm}(\hat{\mathbf{r}})}{r^{l+1}} \tag{7.2}$$

where the multipole moments Q_{lm} are given by

$$Q_{lm} = \int_S \mathrm{d}^3\mathbf{r} Y_{lm}^*(\hat{\mathbf{r}}) r^l \rho(\mathbf{r}) \tag{7.3}$$

From these two equations we notice that the shape of the potential outside the sphere does not at all depend on the precise form of the charge-density. The quantities governing the potential outside the sphere are the multipole moments, and there is an infinite number of charge-densities with the same multipole moments. We will use this property, the arbitrariness of the charge-density inside the sphere, to replace the actual charge-density in the sphere with a more convenient one for our purposes.

What we mean by more convenient we shall see later. First let us examine the solution to (7.1) in the case that the charge-density is given by a plane wave expansion. In general, if a charge-density looks like:

$$\rho(\mathbf{r}) = \sum_{\mathbf{g}} \rho_{\mathbf{g}} e^{i\mathbf{g}\,\mathbf{r}} \tag{7.4}$$

the solution to (7.1) is given by

$$V(\mathbf{r}) = \sum_{\mathbf{g}\neq 0} \frac{4\pi\rho_{\mathbf{g}}}{|\mathbf{g}|^2} e^{i\mathbf{g}\,\mathbf{r}} \tag{7.5}$$

if the system is charge-neutral ($\rho_0 = 0$), and if the system is extended so that no solutions of Laplace's equation contribute.
We shall now approach the solution to (7.1) in two steps:

1. Obtain the potential in the interstitial region.

2. Solve the boundary value problem inside the muffin-tin sphere.

7.2.1 The pseudocharge-density

To obtain the potential in the interstitial, we shall now make use of the arbitrariness of the charge-density in the muffin-tin sphere. Since the potential in the interstitial depends only on the interstitial charge-density and the multipole moments in the sphere, we can replace the real charge-density in the muffin-tin with a *pseudocharge-density*. This pseudocharge-density has to have the same multipole moments as the real charge-density. We know that our original charge-density can be written as (see section 6.3.2):

$$\rho(\mathbf{r}) = \sum_{\mathbf{g}} \rho_{\mathbf{g}} e^{i\mathbf{g}\,\mathbf{r}} + \sum_{L\alpha}^{n_l} \rho_{L\alpha}(r_\alpha) Y_L(\hat{\mathbf{r}_\alpha}) \tag{7.6}$$

We can now write the pseudocharge-density as:

$$\tilde{\rho}(\mathbf{r}) = \sum_{\mathbf{g}} \rho_{\mathbf{g}} e^{i\mathbf{g}\,\mathbf{r}} + \sum_{\alpha} \tilde{\rho}_{\mathbf{g},\alpha} e^{i\mathbf{g}\,\mathbf{r}} \theta(\mathbf{r} \in \text{muffin-tin } \alpha) \tag{7.7}$$

Let us assume we can write the pseudocharge-density as a rapidly converging Fourier series (see next section):

$$\tilde{\rho}(\mathbf{r}) = \sum_{\mathbf{g}} \tilde{\rho}_{\mathbf{g}} e^{i\mathbf{g}\,\mathbf{r}} \tag{7.8}$$

Then we can use (7.5) to find the potential in the interstitial:

$$V_{\text{int}}(\mathbf{r}) = \sum_{\mathbf{g}\neq 0} \frac{4\pi\widetilde{\rho_{\mathbf{g}}}}{|\mathbf{g}|^2} e^{i\mathbf{g}\cdot\mathbf{r}} \tag{7.9}$$

This interstitial potential is also valid on the muffin-tin sphere boundaries. Solving (7.1) inside the sphere then becomes a boundary value problem. We shall see how to solve this problem in section 7.2.3. First let us examine how to find the pseudocharge-density inside the muffin-tin spheres.

7.2.2 The Fourier representation of the pseudocharge-density

In this section I shall show how to obtain the plane wave expansion of the pseudocharge-density. Because of the use of the new LAPW's in the GROW module, the charge-density in our method already has a plane wave part in all space. This means that the step of extending the interstitial charge-density expansion into the spheres as discussed by Weinert[10] is not needed. However as we shall see in section 7.3 this leads to some non-trivial problems when we want to use subvolumes.

The multipoles which we need for our further analysis are the multipoles of the charge-density in the muffin-tin that are not in the plane wave part. In other words we need the multipoles of the radial expansion of the charge-density within the muffin-tin. However, to facilitate the analysis later on when we have subvolumes, I shall write these multipoles as the multipoles of the total charge-density minus the multipoles of the plane wave charge-density in the muffin-tin:

$$\widetilde{Q}^{\alpha}_{lm} = Q^{\alpha}_{lm} - Q^{\text{int},\alpha}_{lm} \tag{7.10}$$

where $Q^{\text{int},\alpha}_{lm}$ is the multipole of the interstitial charge-density extended into the muffin-tin (as is normal in our method). This multipole can be expressed analytically:

$$Q^{\text{int},\alpha}_{lm} = \frac{\sqrt{4\pi}}{3} R^3_{\alpha} \rho^{\text{int}}_{\mathbf{g}=0}\delta_{l0} + \sum_{\mathbf{g}\neq 0} 4\pi i^l \rho^{\text{int}}_{\mathbf{g}} R^{l+3}_{\alpha} \frac{j_{l+1}(|\mathbf{g}|R_{\alpha})}{|\mathbf{g}|R_{\alpha}} e^{i\mathbf{g}\,\mathbf{p}_{\alpha}} Y^*_{lm}(\widehat{\mathbf{g}}) \tag{7.11}$$

where R_{α} is the muffin-tin radius of muffin-tin α and $\mathbf{p}_{\alpha}$ is its position. The multipole defined in (7.10) is the multipole we need to consider to construct the pseudocharge within the muffin-tin sphere.

We replace the real radial charge-density within the muffin-tin by a pseudocharge-density in a spherical harmonic representation, in which the radial part is given by a power series in r:

$$\widetilde{\rho_{\alpha}}(\mathbf{r}_{\alpha}) = \sum_{lm} q^{\alpha}_{lm} Y_{lm}(\widehat{\mathbf{r}_{\alpha}}) \sum_{\eta} a_{\eta} r^{\nu_{\eta}}_{\alpha} \tag{7.12}$$

where a_{η} and ν_{η} are parameters for the power series and the q^{α}_{lm} are constants which ensure that the pseudocharge $\widetilde{\rho_{\alpha}}(\mathbf{r}_{\alpha})$ has the right multipole moments. So substituting (7.12) into (7.3) and equating the resulting multipole to $\widetilde{Q}^{\alpha}_{lm}$ we obtain:

$$\widetilde{Q}^{\alpha}_{lm} = \sum_{l'm'} q^{\alpha}_{l'm'} \int_{S_{\alpha}} \mathrm{d}\widehat{\mathbf{r}_{\alpha}} Y^*_{lm}(\widehat{\mathbf{r}_{\alpha}}) Y_{l'm'}(\widehat{\mathbf{r}_{\alpha}}) \int_0^{R_{\alpha}} \mathrm{d}r_{\alpha} \sum_{\eta} a_{\eta} r^{\nu_{\eta}+l+2}_{\alpha} \tag{7.13a}$$

or

$$q_{lm}^{\alpha} = \tilde{Q}_{lm}^{\alpha}\left[\sum_{\eta} a_{\eta}\frac{R_{\alpha}^{l+\nu_{\eta}+3}}{l+\nu_{\eta}+3}\right]^{-1} \tag{7.13b}$$

By choosing a_η and ν_η appropriately, this expression simplifies, and Weinert has shown that the Fourier series for the pseudocharge-density is given by:

$$\tilde{\rho}_{\mathbf{g}}^{s} = \frac{4\pi}{\Omega}\sum_{lm,\alpha}\frac{(-i)^l(2l+2N_l+3)!!}{(2l+1)!!}\frac{j_{l+N_l+1}(|\mathbf{g}|R_\alpha)}{(|\mathbf{g}|R_\alpha)^{N_l+1}}\tilde{Q}_{lm}^{\alpha}e^{-i\mathbf{g}\,\mathbf{p}_\alpha}Y_{lm}(\hat{\mathbf{g}}) \tag{7.14a}$$

where N_l are parameters which can be varied to optimize convergence. For the case $\mathbf{g} = 0$, we obtain:

$$\tilde{\rho}_{\mathbf{g}=0}^{s} = \frac{4\pi}{\Omega}\sum_{\alpha}\tilde{Q}_{00}^{\alpha} \tag{7.14b}$$

where Ω is the volume of the unit cell. The Fourier expansion for the total pseudocharge which we need to obtain the interstitial potential is:

$$\tilde{\rho}(\mathbf{r}) = \sum_{\mathbf{g}}\left[\rho_{\mathbf{g}} + \tilde{\rho}_{\mathbf{g}}^{s}\right]e^{i\mathbf{g}\,\mathbf{r}} \tag{7.15}$$

This means that the interstitial Coulomb potential is given by:

$$V_{\text{int}}(\mathbf{r}) = \sum_{\mathbf{g}\neq 0}\frac{4\pi\left[\rho_{\mathbf{g}} + \tilde{\rho}_{\mathbf{g}}^{s}\right]}{|\mathbf{g}|^2}e^{i\mathbf{g}\,\mathbf{r}} \tag{7.16}$$

We shall eventually use the interstitial potential as a boundary condition on the muffin-tin to find the potential inside the muffin-tins.

In our layered geometry, the interstitial potential does not only consist of the potential given by (7.15), but we must consider the ρ_0 contribution and solutions of Laplace's equation coming from layer boundary conditions. For the moment we shall assume these to be known. Then the total interstitial potential is given by:

$$V_{\text{int}}(\mathbf{r}) = \sum_{\mathbf{g}} V_{\mathbf{g}}e^{i\mathbf{g}\,\mathbf{r}} + \sum_{\mathbf{G}}\left\{V_{\mathbf{G}}^{+}e^{|\mathbf{G}|z} + V_{\mathbf{G}}^{-}e^{-|\mathbf{G}|z}\right\} + V_1 z + V_2 z^2 \tag{7.17}$$

where $V_{\mathbf{g}}$ (for $\mathbf{g} \neq 0$) is derived from (7.15):

$$V_{\mathbf{g}} = \frac{4\pi\left[\rho_{\mathbf{g}} + \tilde{\rho}_{\mathbf{g}}^{s}\right]}{|\mathbf{g}|^2} \tag{7.18}$$

and V_2 is given by:

$$V_2 = -2\pi\left(\rho_0 + \tilde{\rho}_0^{s}\right) \tag{7.19}$$

The other terms ($V_{\mathbf{g}=0}$, $V_{\mathbf{G}}^{\pm}$, and V_1) arising from the boundaries will be discussed later in section 7.3.2.

7.2.3 The solution inside the muffin-tin sphere

Before we can solve for the potential in the muffin-tin, we must rewrite the charge-density as a single spherical harmonic expansion replacing (7.6):

$$\sum_{lm} \rho'_{lm}(r) Y_{lm}(\hat{\mathbf{r}}) = \sum_{\mathbf{g}} \rho_{\mathbf{g}} e^{i\mathbf{g}\cdot\mathbf{r}} + \sum_{L} \rho_L(r) Y_L(\hat{\mathbf{r}}) \tag{7.20}$$

The re-expansion of the plane wave part of the muffin-tin charge-density into a spherical harmonic expansion is performed in subroutine `REEXPRHO` of the `FISH` module.
We write the potential inside the muffin-tin as:

$$V(\mathbf{r}) = \sum_{lm} V_{lm}(r) Y_{lm}(\hat{\mathbf{r}}) \tag{7.21}$$

The coefficients $V_{lm}(r)$ in (7.18) we obtain by solving the radial Poisson equation:

$$\frac{\mathrm{d}}{\mathrm{d}r}\left(r^2 \frac{\mathrm{d}V_{lm}(r)}{\mathrm{d}r}\right) + 4\pi r^2 \rho'_{lm}(r) - l(l+1)V_{lm}(r) = 0 \tag{7.22}$$

This differential equation is solved in the `FISH` code using the Baylis-Peel version of the Numerov method[21]. Eventually we end up with a solution to (7.22), which we shall call $\widetilde{V}_{lm}(r)$. This solution does not yet comply with the boundary conditions of matching onto the interstitial potential. To satisfy the boundary condition we have to add a solution of Laplace's equation to our solution of (7.22), and obtain:

$$V_{lm}(r) = \widetilde{V}_{lm}(r) - \frac{\sqrt{4\pi}Z}{r}\delta_{l,0} + \left(\frac{r}{R_\alpha}\right)^l \left\{ V_{lm}^{\mathrm{int}}(R_\alpha) - \widetilde{V}_{lm}(r) + \frac{\sqrt{4\pi}Z}{r}\delta_{l,0} \right\} \tag{7.23}$$

The second term on the right-hand side of (7.23) is the $-\frac{Z}{r}$ potential coming from the atomic core. From (7.23) we see that when we set $r = R_\alpha$, we get the values $V_{lm}^{\mathrm{int}}(R_\alpha)$ for $V_{lm}(r)$.

Now we can solve for the potential both in the interstitial and in the muffin-tins: this is exactly what we need. In the next section we shall see why partitioning space into subvolumes complicates things rather drastically.

7.3 It's all in the boundary conditions

In chapter 3 we saw that using subvolumes makes our method very effective. The method then scales linearly with the number of atomic layers in the system we calculate. In this section we shall see, however, that the introduction of subvolumes in the case of Poisson's equations leads to some non-trivial boundary condition problems. These boundary conditions arise both on the muffin-tin-interstitial interface and on the subvolume-subvolume interface.

7.3.1 Finding the correct multipoles for a muffin-tin lying in more than one subvolume

The multipoles needed for the pseudocharge-density expansion were rather simple to calculate before we introduced subvolumes. Since the charge-density inside the muffin-tin sphere is the sum of a plane wave expansion part and a spherical harmonic expansion part, we just calculate

Figure 7.1: *Schematic view of a single subvolume, showing the parts of muffin-tins belonging to this subvolume.*

the multipole of each, and use the multipole of the spherical harmonic expansion to construct the pseudocharge expansion.

However, using subvolumes things become more complicated as a sphere can now lie in more than one subvolume. The spherical harmonic expansion inside the sphere is the same, but we shall divide the plane wave expansion into contributions from each subvolume within which the muffin-tin lies. This means that we first construct multipoles of the plane wave expansions for the parts of the muffin-tins belonging to a certain subvolume. These parts of spheres are sketched in figure 7.1. These partial multipoles are constructed in the subroutine `RHOCAP`. We can then take the relevant building blocks from several subvolumes and add them together to construct the plane wave expansion multipole for each muffin-tin sphere. This process of adding these parts up is done in subroutines `QINTLM` and `ADDUP`.

This method of constructing partial multipoles and then adding them up for each sphere enables us to incorporate boundary conditions on the sides of the embedded stack. Since some of the muffin-tin spheres in the outer layers of the embedded stack will stick out into the substrate, we can use partial multipoles that we have read in from the substrate calculation, to complete the multipoles of those spheres.

7.3.2 Matching potentials from one subvolume to the next

Once we have found the plane wave expansion of the potential in the interstitial in each subvolume, we have to match these potentials in amplitude and derivative across the interfaces between subvolumes by adding on solutions of Laplace's equation appropriate to our geometry:

$$\sum_{\mathbf{G}} \left\{ V_{\mathbf{G}}^{+} e^{|\mathbf{G}|z} + V_{\mathbf{G}}^{-} e^{-|\mathbf{G}|z} \right\} + V_1 z + V_2 z^2 \tag{7.24}$$

We have used these terms before (section 7.2.2) when discussing the potential, and have stated that these terms arise from the subvolume boundary conditions. In this section we shall see how we fix the coefficients $V_{\mathbf{g}=0}$, $V_{\mathbf{G}}^{\pm}$, and V_1.

First we write the potential for subvolume I in the following way:

$$V^I(\mathbf{r}) = \sum_{\mathbf{G}} V_{\mathbf{G}}^I(z) e^{i\mathbf{G}\cdot\mathbf{R}} \tag{7.25}$$

where $V_{\mathbf{G}}^I$ is given by:

$$V_{\mathbf{G}}^I(z) = \begin{cases} A^I + B^I z + {\sum_{g_z}}' V_{0,g_z}^I e^{\imath g_z z} & \text{if } \mathbf{G} = 0 \\ V_{\mathbf{G}}^{I,+} e^{|\mathbf{G}|z} + V_{\mathbf{G}}^{I,-} e^{-|\mathbf{G}|z} + \sum_{g_z} V_{\mathbf{G},g_z}^I e^{\imath g_z z} & \text{if } \mathbf{G} \neq 0 \end{cases} \tag{7.26}$$

The prime on the sum indicates that $g_z = 0$ is excluded.

In general, when we have a stack of N subvolumes, continuity of potential and field gives for the $N-1$ internal interfaces (i.e. within the stack):

$$V_{\mathbf{G}}^I(z_R^I) = V_{\mathbf{G}}^{I+1}(z_L^{I+1}) e^{\imath \mathbf{G}\,\mathbf{R}_I^+} \tag{7.27a}$$

$$\frac{\mathrm{d}V_{\mathbf{G}}^I}{\mathrm{d}z}(z_R^I) = \frac{\mathrm{d}V_{\mathbf{G}}^{I+1}}{\mathrm{d}z}(z_L^{I+1}) e^{\imath \mathbf{G}\,\mathbf{R}_I^+} \tag{7.27b}$$

where $\mathbf{R}_I^+$ is the parallel vector connecting the origin of subvolume I to the origin of subvolume $I+1$. However, the matching conditions for the two external interfaces depend on the type of system under investigation.

In general we have three types of system: bulk, interface, and surface. In the following sections I shall discuss each in detail and derive expression for the matching conditions.

7.3.2.1 Boundary conditions for a bulk system

As a first step we shall adopt the following notation:

$$\mathrm{VGL}(I, \mathbf{G}) = \sum_{g_z} V_{\mathbf{G},g_z}^I e^{\imath g_z z_L^I} \tag{7.28a}$$

$$\mathrm{VGR}(I, \mathbf{G}) = \sum_{g_z} V_{\mathbf{G},g_z}^I e^{\imath g_z z_R^I} \tag{7.28b}$$

$$\mathrm{dVGL}(I, \mathbf{G}) = \sum_{g_z} V_{\mathbf{G},g_z}^I g_z e^{\imath g_z z_L^I} \tag{7.28c}$$

$$\mathrm{dVGR}(I, \mathbf{G}) = \sum_{g_z} V_{\mathbf{G},g_z}^I g_z e^{\imath g_z z_R^I} \tag{7.28d}$$

where the $z_{L/R}^I$ indicate the z-positions of the planes separating the subvolumes in terms of the z-origin of subvolume I. Using this notation, let us examine the case of an internal interface.

Each $\mathbf{G}$ (the surface layer-parallel reciprocal lattice vector) can be treated separately, and for $\mathbf{G} = 0$ the amplitude matching condition (7.27a) becomes:

$$A^I + B^I z_R^I + \mathrm{VGR}(I, 0) = A^{I+1} + B^{I+1} z_L^{I+1} + \mathrm{VGL}(I+1, 0) \tag{7.29}$$

and the derivative matching condition (7.27b) becomes:

$$B^I + \mathrm{dVGR}(I, 0) = B^{I+1} + \mathrm{dVGL}(I+1, 0) \tag{7.30}$$

Eventually we want to write down a matrix equation, with the unknown A^I and B^I as a vector. So we rewrite (7.29) and (7.30) in the following way:

$$A^I + B^I z_R^I - A^{I+1} - B^{I+1} z_L^{I+1} = \mathrm{VGL}(I+1,0) - \mathrm{VGR}(I,0) \tag{7.31}$$

$$B^I - B^{I+1} = \mathrm{dVGL}(I+1,0) - \mathrm{dVGR}(I,0) \tag{7.32}$$

The vector containing the unknowns is organized as:

$$x = \begin{pmatrix} A^1 \\ B^1 \\ A^2 \\ B^2 \\ \vdots \\ A^N \\ B^N \end{pmatrix} \tag{7.33}$$

We write the matrix equation as:

$$\mathbb{A}x = y \tag{7.34}$$

From (7.31) and (7.32) we get the following elements of $\mathbb{A}$ and y:

$$\mathbb{A}_{\jmath,\jmath} = 1 \tag{7.35a}$$

$$\mathbb{A}_{\jmath,\jmath+1} = z_R^I \tag{7.35b}$$

$$\mathbb{A}_{\jmath,\jmath+2} = -1 \tag{7.35c}$$

$$\mathbb{A}_{\jmath,\jmath+3} = -z_L^{I+1} \tag{7.35d}$$

$$y_\jmath = \mathrm{VGL}(I+1,0) - \mathrm{VGR}(I,0) \tag{7.35e}$$

$$\mathbb{A}_{\jmath+1,\jmath} = 0 \tag{7.35f}$$

$$\mathbb{A}_{\jmath+1,\jmath+1} = 1 \tag{7.35g}$$

$$\mathbb{A}_{\jmath+1,\jmath+2} = 0 \tag{7.35h}$$

$$\mathbb{A}_{\jmath+1,\jmath+3} = -1 \tag{7.35i}$$

$$y_{\jmath+1} = \mathrm{dVGL}(I+1,0) - \mathrm{dVGR}(I,0) \tag{7.35j}$$

where $\jmath = 2 * I - 1$. For a bulk system we have the following extra boundary conditions on the $\mathbf{G} = 0$ term for the external interfaces:

$$V_0^1(z_L^1) = \text{constant} \tag{7.36a}$$

$$V_0^N(z_R^N) = \text{constant} \tag{7.36b}$$

If we examine (7.36a) and (7.36b), we get:

$$A^1 + B^1 z_L^1 + \mathrm{VGL}(1,0) = \text{constant} \tag{7.37a}$$

$$A^N + B^N z_R^N + \mathrm{VGR}(N,0) = \text{constant} \tag{7.37b}$$

which lead to the following matrix-elements:

$$\mathbb{A}_{j,1} = 1 \tag{7.38a}$$
$$\mathbb{A}_{j,2} = z_L^1 \tag{7.38b}$$
$$y_j = \text{constant} - \text{VGL}(1,0) \tag{7.38c}$$
$$\mathbb{A}_{j+1,j} = 1 \tag{7.38d}$$
$$\mathbb{A}_{j+1,j+1} = z_R^N \tag{7.38e}$$
$$y_{j+1} = \text{constant} - \text{VGR}(N,0) \tag{7.38f}$$

where $j = 2 * N - 1$.

We can now solve for the A^I and B^I by inverting $\mathbb{A}$ and writing:

$$x = \mathbb{A}^{-1} y \tag{7.39}$$

In this way we obtain those A^I and B^I for which the $\mathbf{G} = 0$ component of the potential is continuous in derivative and amplitude over the internal interfaces, is continuous over the external interfaces, and has the periodicity of the bulk repeat unit (BRU).

For finite $\mathbf{G}$, things become a bit more complex. Now (7.27a) and (7.27b) become:

$$V_{\mathbf{G}}^{I,+} e^{|\mathbf{G}|z_R^I} + V_{\mathbf{G}}^{I,-} e^{-|\mathbf{G}|z_R^I} + \text{VGR}(I,\mathbf{G}) = \left(V_{\mathbf{G}}^{I+1,+} e^{|\mathbf{G}|z_L^{I+1}} + V_{\mathbf{G}}^{I+1,-} e^{-|\mathbf{G}|z_L^{I+1}} + \text{VGL}(I+1,\mathbf{G})\right) e^{i\mathbf{G}\,\mathbf{R}_I^+} \tag{7.40a}$$

$$V_{\mathbf{G}}^{I,+}|\mathbf{G}|e^{|\mathbf{G}|z_R^I} - V_{\mathbf{G}}^{I,-}|\mathbf{G}|e^{-|\mathbf{G}|z_R^I} + \text{dVGR}(I,\mathbf{G}) = \left(V_{\mathbf{G}}^{I+1,+}|\mathbf{G}|e^{|\mathbf{G}|z_L^{I+1}} - V_{\mathbf{G}}^{I+1,-}|\mathbf{G}|e^{-|\mathbf{G}|z_L^{I+1}} + \text{dVGL}(I+1,\mathbf{G})\right) e^{i\mathbf{G}\cdot\mathbf{R}_I^+} \tag{7.40b}$$

In preparation for the matrix equation we rewrite this as:

$$V_{\mathbf{G}}^{I,+} e^{|\mathbf{G}|z_R^I} + V_{\mathbf{G}}^{I,-} e^{-|\mathbf{G}|z_R^I} - V_{\mathbf{G}}^{I+1,+} e^{|\mathbf{G}|z_L^{I+1}} e^{i\mathbf{G}\,\mathbf{R}_I^+} - V_{\mathbf{G}}^{I+1,-} e^{-|\mathbf{G}|z_L^{I+1}} e^{i\mathbf{G}\,\mathbf{R}_I^+} = \text{VGL}(I+1,\mathbf{G}) e^{i\mathbf{G}\,\mathbf{R}_I^+} - \text{VGR}(I,\mathbf{G}) \tag{7.41a}$$

$$V_{\mathbf{G}}^{I,+}|\mathbf{G}|e^{|\mathbf{G}|z_R^I} - V_{\mathbf{G}}^{I,-}|\mathbf{G}|e^{-|\mathbf{G}|z_R^I} - V_{\mathbf{G}}^{I+1,+}|\mathbf{G}|e^{|\mathbf{G}|z_L^{I+1}} e^{i\mathbf{G}\,\mathbf{R}_I^+} + V_{\mathbf{G}}^{I+1,-}|\mathbf{G}|e^{-|\mathbf{G}|z_L^{I+1}} e^{i\mathbf{G}\,\mathbf{R}_I^+} = \text{dVGL}(I+1,\mathbf{G}) e^{i\mathbf{G}\,\mathbf{R}_I^+} - \text{dVGR}(I,\mathbf{G}) \tag{7.41b}$$

The vector containing the unknowns is now organized as:

$$x = \begin{pmatrix} V_{\mathbf{G}}^{1,+} \\ V_{\mathbf{G}}^{1,-} \\ V_{\mathbf{G}}^{2,+} \\ V_{\mathbf{G}}^{2,-} \\ \vdots \\ V_{\mathbf{G}}^{N,+} \\ V_{\mathbf{G}}^{N,-} \end{pmatrix} \tag{7.42}$$

and again we write the matrix equation as:

$$\mathbb{A}x = y \tag{7.43}$$

From (7.41a) and (7.41b) we find the following elements of $\mathbb{A}$ and y:

$$\mathbb{A}_{j,j} = e^{|\mathbf{G}|z_R^I} \tag{7.44a}$$

$$\mathbb{A}_{j,j+1} = e^{-|\mathbf{G}|z_R^I} \tag{7.44b}$$

$$\mathbb{A}_{j,j+2} = -e^{|\mathbf{G}|z_L^{I+1}} e^{i\mathbf{G}\,\mathbf{R}_I^+} \tag{7.44c}$$

$$\mathbb{A}_{j,j+3} = -e^{-|\mathbf{G}|z_L^{I+1}} e^{i\mathbf{G}\,\mathbf{R}_I^+} \tag{7.44d}$$

$$y_j = \mathrm{VGL}(I+1,\mathbf{G}) e^{i\mathbf{G}\cdot\mathbf{R}_I^+} - \mathrm{VGR}(I,\mathbf{G}) \tag{7.44e}$$

$$\mathbb{A}_{j+1,j} = |\mathbf{G}| e^{|\mathbf{G}|z_R^I} \tag{7.44f}$$

$$\mathbb{A}_{j+1,j+1} = -|\mathbf{G}| e^{-|\mathbf{G}|z_R^I} \tag{7.44g}$$

$$\mathbb{A}_{j+1,j+2} = -|\mathbf{G}| e^{|\mathbf{G}|z_L^{I+1}} e^{i\mathbf{G}\,\mathbf{R}_I^+} \tag{7.44h}$$

$$\mathbb{A}_{j+1,j+3} = |\mathbf{G}| e^{-|\mathbf{G}|z_L^{I+1}} e^{i\mathbf{G}\,\mathbf{R}_I^+} \tag{7.44i}$$

$$y_{j+1} = \mathrm{dVGL}(I+1,\mathbf{G}) e^{i\mathbf{G}\cdot\mathbf{R}_I^+} - \mathrm{dVGR}(I,\mathbf{G}) \tag{7.44j}$$

where $j = 2 * I - 1$. Again we have bulk boundary conditions for the external interfaces:

$$V_{\mathbf{G}}^N(z_R^N) = V_{\mathbf{G}}^1(z_L^1) e^{i\mathbf{G}\cdot\mathbf{R}_{1,N}} \tag{7.45a}$$

$$\frac{\mathrm{d}V_{\mathbf{G}}^N}{\mathrm{d}z}(z_R^N) = \frac{\mathrm{d}V_{\mathbf{G}}^1}{\mathrm{d}z}(z_L^1) e^{i\mathbf{G}\,\mathbf{R}_{1,N}} \tag{7.45b}$$

These become in the notation defined in (7.28):

$$V_{\mathbf{G}}^{N,+} e^{|\mathbf{G}|z_R^N} + V_{\mathbf{G}}^{N,-} e^{-|\mathbf{G}|z_R^N} + \mathrm{VGR}(N,\mathbf{G}) = \left(V_{\mathbf{G}}^{1,+} e^{|\mathbf{G}|z_L^1} + V_{\mathbf{G}}^{1,-} e^{-|\mathbf{G}|z_L^1} + \mathrm{VGL}(1,\mathbf{G})\right) e^{i\mathbf{G}\,\mathbf{R}_{1,N}} \tag{7.46a}$$

$$V_{\mathbf{G}}^{N,+} |\mathbf{G}| e^{|\mathbf{G}|z_R^N} - V_{\mathbf{G}}^{N,-} |\mathbf{G}| e^{-|\mathbf{G}|z_R^N} + \mathrm{dVGR}(N,\mathbf{G}) = \left(V_{\mathbf{G}}^{1,+} |\mathbf{G}| e^{|\mathbf{G}|z_L^1} - V_{\mathbf{G}}^{1,-} |\mathbf{G}| e^{-|\mathbf{G}|z_L^1} + \mathrm{dVGL}(1,\mathbf{G})\right) e^{i\mathbf{G}\,\mathbf{R}_{1,N}} \tag{7.46b}$$

We rewrite these as:

$$V_{\mathbf{G}}^{N,+} e^{|\mathbf{G}|z_R^N} + V_{\mathbf{G}}^{N,-} e^{-|\mathbf{G}|z_R^N} - V_{\mathbf{G}}^{1,+} e^{|\mathbf{G}|z_L^1} e^{i\mathbf{G}\,\mathbf{R}_{1,N}} - V_{\mathbf{G}}^{1,-} e^{-|\mathbf{G}|z_L^1} e^{i\mathbf{G}\,\mathbf{R}_{1,N}} = \mathrm{VGL}(1,\mathbf{G}) e^{i\mathbf{G}\,\mathbf{R}_{1,N}} - \mathrm{VGR}(N,\mathbf{G}) \tag{7.47a}$$

$$V_{\mathbf{G}}^{N,+} |\mathbf{G}| e^{|\mathbf{G}|z_R^N} - V_{\mathbf{G}}^{N,-} |\mathbf{G}| e^{-|\mathbf{G}|z_R^N} - V_{\mathbf{G}}^{1,+} |\mathbf{G}| e^{|\mathbf{G}|z_L^1} e^{i\mathbf{G}\,\mathbf{R}_{1,N}} + V_{\mathbf{G}}^{1,-} |\mathbf{G}| e^{-|\mathbf{G}|z_L^1} e^{i\mathbf{G}\,\mathbf{R}_{1,N}} = \mathrm{dVGL}(1,\mathbf{G}) e^{i\mathbf{G}\,\mathbf{R}_{1,N}} - \mathrm{dVGR}(N,\mathbf{G}) \tag{7.47b}$$

This leads to the following elements of $\mathbb{A}$ and y:

$$\mathbb{A}_{j,1} = -e^{|\mathbf{G}|z_L^1} e^{i\mathbf{G}\,\mathbf{R}_{1,N}} \tag{7.48a}$$

$$\mathbb{A}_{j,2} = -e^{-|\mathbf{G}|z_L^1} e^{i\mathbf{G}\,\mathbf{R}_{1,N}} \tag{7.48b}$$

$$\mathbb{A}_{j,j} = e^{|\mathbf{G}|z_R^N} \tag{7.48c}$$

$$\mathbb{A}_{j,j+1} = e^{-|\mathbf{G}|z_R^N} \tag{7.48d}$$

$$y_j = \mathrm{VGL}(1,\mathbf{G}) e^{i\mathbf{G}\cdot\mathbf{R}_{1,N}} - \mathrm{VGR}(N,\mathbf{G}) \tag{7.48e}$$

$$\mathbb{A}_{j+1,1} = -|\mathbf{G}| e^{|\mathbf{G}|z_L^1} e^{i\mathbf{G}\,\mathbf{R}_{1,N}} \tag{7.48f}$$

$$\mathbb{A}_{j+1,2} = |\mathbf{G}| e^{-|\mathbf{G}|z_L^1} e^{i\mathbf{G}\,\mathbf{R}_{1,N}} \tag{7.48g}$$

$$\mathbb{A}_{j+1,j} = |\mathbf{G}| e^{|\mathbf{G}|z_R^N} \tag{7.48h}$$

$$\mathbb{A}_{j+1,j+1} = -|\mathbf{G}| e^{-|\mathbf{G}|z_R^N} \tag{7.48i}$$

$$y_{j+1} = \mathrm{dVGL}(1,\mathbf{G}) e^{i\mathbf{G}\,\mathbf{R}_{1,N}} - \mathrm{dVGR}(N,\mathbf{G}) \tag{7.48j}$$

where $j = 2 * N - 1$.

So for every $\mathbf{G}$ we have a matrix $\mathbb{A}$ and a vector y. By inverting the matrix $\mathbb{A}$ we can calculate the coefficients needed to match the potential on all interfaces and make the potential periodic with the bulk repeat unit.

In the FISH module, the job of calculating the quantities defined in (7.28) is performed in the subroutine POTSV. The matching and subsequent calculation of the matching coefficients is done in BULKBC for bulk systems. For non-bulk systems, the code calls the subroutine POTBC. The expressions used in this subroutine will be derived in the following two sections.

7.3.2.2 Boundary conditions for an interface system

For the interface system, again we first examine the $\mathbf{G} = 0$ term. We see that for internal interfaces nothing changes, so (7.35a)-(7.35j) still hold. On the external interfaces we have to match to the potential of the semi-infinite substrates. To do this, the boundary potentials for the semi-infinite substrate are read in, which we shall call $V_{0,L}^{\infty}$ and $V_{0,R}^{\infty}$. So we now obtain the following relationship between the $\mathbf{G} = 0$ terms in subvolumes 1 and N and the read-in substrate boundary values:

$$A^1 + B^1 z_L^1 + \mathrm{VGL}(1,0) = V_{0,L}^{\infty} \tag{7.49a}$$

$$A^N + B^N z_R^N + \mathrm{VGR}(N,0) = V_{0,R}^{\infty} \tag{7.49b}$$

leading to the following matrix-elements:

$$\mathbb{A}_{j,1} = 1 \tag{7.50a}$$

$$\mathbb{A}_{j,2} = z_L^1 \tag{7.50b}$$

$$y_j = V_{0,L}^{\infty} - \mathrm{VGL}(1,0) \tag{7.50c}$$

$$\mathbb{A}_{j+1,j} = 1 \tag{7.50d}$$

$$\mathbb{A}_{j+1,j+1} = z_R^N \tag{7.50e}$$

$$y_{j+1} = V_{0,R}^{\infty} - \mathrm{VGR}(N,0) \tag{7.50f}$$

where $j = 2 * N - 1$. These enable us to find the A^I and B^I which solve the matching conditions.

For finite $\mathbf{G}$ we also match the amplitude of the potential to the values given by the semi-infinite substrate, $V^{\infty}_{\mathbf{G},L}$ and $V^{\infty}_{\mathbf{G},R}$ (again internal interfaces stay the same as in (7.44a)-(7.44j)). So we have:

$$V^1_{\mathbf{G}}(z^1_L) = V^{\infty}_{\mathbf{G},L} \tag{7.51a}$$

$$V^N_{\mathbf{G}}(z^N_R) = V^{\infty}_{\mathbf{G},R} \tag{7.51b}$$

or, written out in full:

$$V^{1,+}_{\mathbf{G}} e^{|\mathbf{G}|z^1_L} + V^{1,-}_{\mathbf{G}} e^{-|\mathbf{G}|z^1_L} + \mathrm{VGL}(1,\mathbf{G}) = V^{\infty}_{\mathbf{G},L} \tag{7.52a}$$

$$V^{N,+}_{\mathbf{G}} e^{|\mathbf{G}|z^N_R} + V^{N,-}_{\mathbf{G}} e^{-|\mathbf{G}|z^N_R} + \mathrm{VGR}(N,\mathbf{G}) = V^{\infty}_{\mathbf{G},R} \tag{7.52b}$$

This leads to the following matrix-elements:

$$\mathbb{A}_{j,1} = e^{|\mathbf{G}|z^1_L} \tag{7.53a}$$

$$\mathbb{A}_{j,2} = e^{-|\mathbf{G}|z^1_L} \tag{7.53b}$$

$$y_j = V^{\infty}_{\mathbf{G},L} - \mathrm{VGL}(1,\mathbf{G}) \tag{7.53c}$$

$$\mathbb{A}_{j+1,j} = e^{|\mathbf{G}|z^N_R} \tag{7.53d}$$

$$\mathbb{A}_{j+1,j+1} = e^{-|\mathbf{G}|z^N_R} \tag{7.53e}$$

$$y_{j+1} = V^{\infty}_{\mathbf{G},R} - \mathrm{VGR}(N,\mathbf{G}) \tag{7.53f}$$

where $j = 2 * N - 1$.

Using the inverse of the matrix, we can now find the coefficients that solve the matching conditions to obtain a potential that is internally matched in amplitude and derivative and which is matched in amplitude to the potentials in the semi-infinite substrates.

7.3.2.3 Boundary conditions for a surface system

For the surface system, the external boundary conditions are different again. On the side of the bulk, the boundary condition is the same as for the interface system. On the vacuum side it is not the amplitude of the potential which is fixed to an outside value, but the derivative. The physical requirement that there is zero field in the vacuum means that the potential must have zero derivative in the vacuum, and for $\mathbf{G} = 0$ the external boundary conditions become:

$$A^1 + B^1 z^1_L + \mathrm{VGL}(1,0) = V^{\infty}_{0,L} \tag{7.54a}$$

$$B^N + \mathrm{dVGR}(N,0) = 0 \tag{7.54b}$$

This leads to the following matrix-elements:

$$\mathbb{A}_{j,1} = 1 \tag{7.55a}$$

$$\mathbb{A}_{j,2} = z^1_L \tag{7.55b}$$

$$y_j = V^{\infty}_{0,L} - \mathrm{VGL}(1,0) \tag{7.55c}$$

$$\mathbb{A}_{j+1,j} = 0 \tag{7.55d}$$

$$\mathbb{A}_{j+1,j+1} = 1 \tag{7.55e}$$

$$y_{j+1} = -\mathrm{dVGR}(N,0) \tag{7.55f}$$

where $j = 2*N-1$. We subsequently solve for A^I and B^I - note that matching the derivative allows the amplitude of the potential, hence the work function, to find its own value.
For finite $\mathbf{G}$ we get:

$$V_{\mathbf{G}}^{1,+}e^{|\mathbf{G}|z_L^1} + V_{\mathbf{G}}^{1,-}e^{-|\mathbf{G}|z_L^1} + \mathrm{VGL}(1,\mathbf{G}) = V_{\mathbf{G},L}^{\infty} \tag{7.56a}$$

$$V_{\mathbf{G}}^{N,+}|\mathbf{G}|e^{|\mathbf{G}|z_R^N} - V_{\mathbf{G}}^{N,-}|\mathbf{G}|e^{-|\mathbf{G}|z_R^N} + \mathrm{dVGR}(N,\mathbf{G}) = 0 \tag{7.56b}$$

which lead to the following matrix-elements:

$$\mathbb{A}_{j,1} = e^{|\mathbf{G}|z_L^1} \tag{7.57a}$$

$$\mathbb{A}_{j,2} = e^{-|\mathbf{G}|z_L^1} \tag{7.57b}$$

$$y_j = V_{\mathbf{G},L}^{\infty} - \mathrm{VGL}(1,\mathbf{G}) \tag{7.57c}$$

$$\mathbb{A}_{j+1,j} = |\mathbf{G}|e^{|\mathbf{G}|z_R^N} \tag{7.57d}$$

$$\mathbb{A}_{j+1,j+1} = -|\mathbf{G}|e^{-|\mathbf{G}|z_R^N} \tag{7.57e}$$

$$y_{j+1} = -\mathrm{dVGR}(N,\mathbf{G}) \tag{7.57f}$$

where $j = 2*N-1$. Solving for the matching coefficients once again gives us a potential that is internally matched, matches in amplitude to the bulk and has zero derivative (i.e. zero field) in the vacuum.

We have now seen how we can make the interstitial potential in all subvolumes to be continuous and to match to the external boundary conditions. The only remaining step in finding the electrostatic potential is to find the potential in the muffin-tins as described in section 7.2.3. For this we need the boundary conditions on the muffin-tin sphere. These are discussed in the next section.

7.3.3 Boundary conditions on a sphere intersecting multiple subvolumes

In section 7.2.3 we discussed how to solve Poisson's equation inside the muffin-tin. We mentioned there that to be able to match the solution to the interstitial potential, we need the correct boundary condition on the muffin-tin sphere. This boundary condition is a spherical harmonic expansion of the interstitial potential on the muffin-tin sphere. However, as can be seen in figure 7.2, when the muffin-tin lies in more than one subvolume, there isn't just one interstitial potential which we can use for the boundary condition. Let us call the interstitial potential in the middle subvolume $V_{\mathrm{int}}^I(\mathbf{r})$, and the ones in the neighbouring subvolumes $V_{\mathrm{int}}^{I-1}(\mathbf{r})$ and $V_{\mathrm{int}}^{I+1}(\mathbf{r})$. Now let us take an extra look at the expression for the spherical harmonics expansion coefficients on the muffin-tin:

$$V_{lm}^{\mathrm{int}}(R_\alpha) = \int_{S_\alpha} \mathrm{d}\widehat{\mathbf{r}_\alpha} Y_{lm}^*(\widehat{\mathbf{r}_\alpha}) V_{\mathrm{int}}(\mathbf{r}_\alpha) \tag{7.58}$$

In this case the integral breaks down into three separate integrals:

$$V_{lm}^{\mathrm{int}}(R_\alpha) = \int_{\substack{\text{left}\\ \text{cap}}} \mathrm{d}\widehat{\mathbf{r}_\alpha} Y_{lm}^*(\widehat{\mathbf{r}_\alpha}) V_{\mathrm{int}}^{I-1}(\mathbf{r}_\alpha) + \int_{\substack{\text{sphere}\\ \text{segment}}} \mathrm{d}\widehat{\mathbf{r}_\alpha} Y_{lm}^*(\widehat{\mathbf{r}_\alpha}) V_{\mathrm{int}}^{I}(\mathbf{r}_\alpha) + \int_{\substack{\text{right}\\ \text{cap}}} \mathrm{d}\widehat{\mathbf{r}_\alpha} Y_{lm}^*(\widehat{\mathbf{r}_\alpha}) V_{\mathrm{int}}^{I+1}(\mathbf{r}_\alpha) \tag{7.59}$$

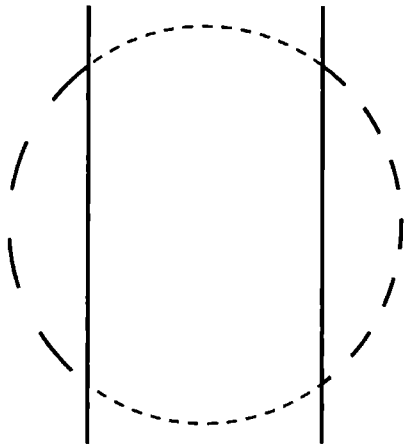


Figure 7.2: *Muffin-tin lying in three subvolumes. The different dashes on the muffin-tin sphere indicate the different boundary conditions.*

Now when we look at the expressions for the interstitial potential (see (7.17)), we notice that we need integrals of the following types:

$$\begin{aligned} &\int \mathrm{d}\widehat{\mathbf{r}_\alpha} Y_{lm}^*(\widehat{\mathbf{r}_\alpha}) z \qquad \int \mathrm{d}\widehat{\mathbf{r}_\alpha} Y_{lm}^*(\widehat{\mathbf{r}_\alpha}) z^2 \\ &\int \mathrm{d}\widehat{\mathbf{r}_\alpha} Y_{lm}^*(\widehat{\mathbf{r}_\alpha}) e^{i\mathbf{g}\cdot\mathbf{r}_\alpha} \quad \int \mathrm{d}\widehat{\mathbf{r}_\alpha} Y_{lm}^*(\widehat{\mathbf{r}_\alpha}) e^{\pm|\mathbf{G}|z} \end{aligned} \tag{7.60}$$

All these integrals have to be calculated over partial spheres, either caps or spheres with caps subtracted. In this thesis I shall not go into the details of calculating these integrals. They are calculated in the subroutines `VCAP` and `VSPHERE`.

7.4 Exchange-correlation potential

Having determined the electrostatic potential we turn now to the exchange-correlation potential of density functional theory (DFT). This is evaluated within the `FISH` module using the local density approximation (LDA). Six expressions for the local density exchange-correlation potential (and energy) are incorporated into the code:

1. Kohn-Sham exchange[22].
2. X_α method[22].
3. Wigner interpolation[23].
4. Hedin-Lundqvist[24].
5. Von Barth-Hedin[25].
6. Ceperley-Alder[26].

All of these methods are incorporated in the subroutine EXCORR. The input variable lxc selects one from the above list (e.g. lxc=4 means: use Hedin-Lundqvist).

This subroutine only returns a potential value, given a charge-density value. What we want in fact is an expansion of the exchange-correlation potential of the form:

$$V_{xc}(\mathbf{r}) = \sum_{\mathbf{g}} V_{\mathbf{g}}^{xc} e^{i\mathbf{g}\cdot\mathbf{r}} \theta_{\rm int}(\mathbf{r}) + \sum_{lm} V_{lm}^{xc}(r) Y_{lm}(\hat{\mathbf{r}}) \theta_{\rm mt}(\mathbf{r}) \tag{7.61}$$

where the function $\theta_{\rm int}(\mathbf{r})$ is one in the interstitial and zero elsewhere, and the function $\theta_{\rm mt}(\mathbf{r})$ is one within a muffin-tin and zero in the interstitial. As the charge-density is continuous, the exchange-correlation potential constructed from the charge-density will be also continuous.

7.4.1 The exchange-correlation potential in the interstitial

We shall first concentrate on the interstitial exchange-correlation potential. We solve for the coefficients $V_{\mathbf{g}}^{xc}$ by a least-squares-fit-like method. Suppose we construct a matrix which looks like this:

$$\mathbb{B}_{i,\mathbf{g}} = e^{i\mathbf{g}\cdot\mathbf{r}_i} \tag{7.62}$$

We can then write down the following matrix equation:

$$\mathbf{v}_{xc} = \mathbb{B}\mathbf{V}^{xc} \tag{7.63}$$

where $\mathbf{v}_{xc}$ is the vector containing the exchange-correlation potential $V(\mathbf{r}_i)$ for a selection of random points in the interstitial, and $\mathbf{V}^{xc}$ is the vector of coefficients $V_{\mathbf{g}}^{xc}$. Normally, to solve for the vector $\mathbf{V}^{xc}$, we would invert the matrix $\mathbb{B}$ and multiply the inverse with the vector $\mathbf{v}_{xc}$. However, in general, the matrix $\mathbb{B}$ is not square and thus has no simple inverse. A method for solving (7.63) exists and is called Singular Value Decomposition (SVD)[27]. This is what is used in the code.

The points $\mathbf{r}_i$ are chosen randomly in the interstitial of the subvolume, and at each point the charge-density is evaluated. This charge-density is then fed to EXCORR to return the exchange-correlation potential at that point from which the vector $\mathbf{v}_{xc}$ can be constructed. The matrix $\mathbb{B}$ is fairly simple to construct once we know the random points. Then a call to a SVD routine is made and the coefficients $V_{\mathbf{g}}^{xc}$ are determined.

Since the SVD algorithm is quite expensive ($\approx N^3$, where N is the number of $\mathbf{g}$ vectors), we use the symmetry property that each three-dimensional star of $\mathbf{g}$-vectors has a single $V_{\mathbf{g}}^{xc}$. (The three-dimensional star of $\mathbf{g}$-vectors is defined as those $\mathbf{g}$ related by the point-group of the system under investigation.) We can also limit the effect of the exchange-correlation to a certain maximum length of $\mathbf{g}$, assuming that for the larger $\mathbf{g}$ the kinetic energy will eventually dominate any potential.

7.4.2 The exchange-correlation potential in the muffin-tin

For finding the expansion coefficients inside the muffin-tin we follow a similar strategy. Since we want the exchange-correlation potential on a grid, we solve for every r separately. For every r we choose a number of random angles, from which we construct the matrix $\mathbb{B}$ which now looks like:

$$\mathbb{B}_{i,lm} = Y_{lm}(\hat{\mathbf{r}}_i) \tag{7.64}$$

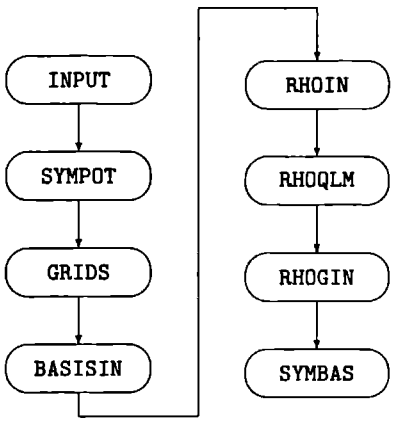


Figure 7 3 *Small scale structure of input and initialisation part of the* FISH *module*

Again we get a matrix-equation as in (7 63), in which the vector $\mathbf{v}_{xc}$ consists of the exchange-correlation potential values at the random points given by r and the random angles The vector $\mathbf{V}^{xc}$ now consists of the expansion coefficients $V_{lm}^{xc}(r)$ Again we solve (7 64) using SVD, but this time the matrix $\mathbb{B}$ is relatively small and the SVD is not the time-limiting step That is the reason that we can easily perform this procedure for every r on the radial grid

This means that we now have both the muffin-tin and the interstitial expansion coefficients and therefore the entire exchange-correlation potential This is added to the electrostatic potential, to create the total potential, to be used as input for the GROW module

7.5 The structure of the FISH module

As in the previous chapter on the GROW module we shall now look in a little more detail at the structure of the FISH module The overall structure of the FISH module is the same as for GROW the first part handles initialisation and input, the second part performs the calculation and the third parts handles the output

7.5.1 Input and initialisation

In figure 7 3 we see a flow chart of the input and initialisation part of the FISH module Let us now examine each part in detail

- INPUT This subroutine has more or less the same function as the NEWIN routine in the GROW module The geometry of the problem is input, as well as several control parameters

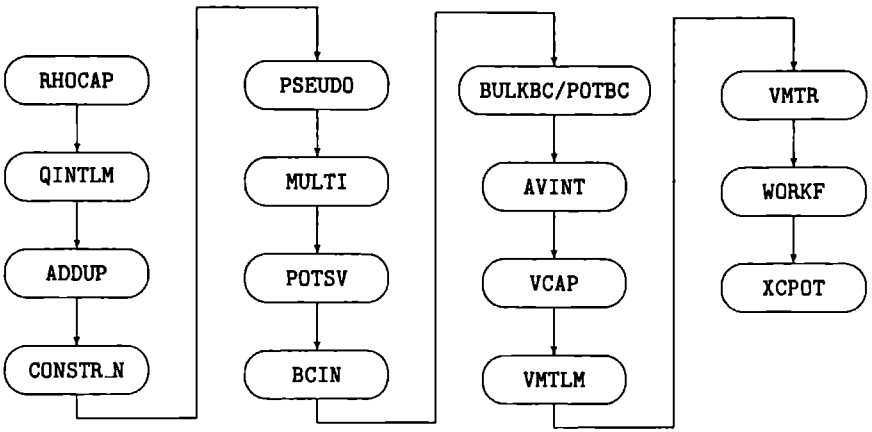


Figure 7.4: *Small scale structure of calculation part of the* FISH *module.*

- SYMPOT: The information input in the previous subroutine is used to find out which (l, m) components of the potential are allowed to be non-zero by symmetry arguments.
- GRIDS: The basis vectors of the lattice are checked to ensure a right-hand coordinate system.
- BASISIN: The **g** basis of the charge-density is read in. This basis was output by BASISOUT in the module GROW.
- RHOIN: This routine reads in the muffin-tin charge-density expansion. This expansion was output by the routine RHOOUT of the module GROW.
- RHOQLM: The multipole moments of the charge-density are read in. Originally these moments were written to file by the routine GETMULTIP of the GROW module.
- RHOGIN: Of course we also need the interstitial (i.e. plane wave) charge-density. This quantity is input by RHOGIN.
- SYMBAS: The basis for the potential is symmetrized and put into three-dimensional stars, to be used in other routines, e.g. in the routine XCPOT which calculates the exchange-correlation potential (see section 7.4.1).

At this stage in the code everything is set up and the actual calculation of the potential can begin.

7.5.2 Calculating the potential

The structure of the FISH module is not as complicated as the GROW module. In figure 7.4 a flow-chart is shown coupling the different subroutines:

- RHOCAP: The partial multipoles in caps and sphere segments are calculated in this routine.
- QINTLM: The partial multipoles are prepared for the pseudocharge expansion and addition.
- ADDUP: This subroutine performs the addition of the partial multipoles in caps and sphere segments. For the two outer subvolumes, it also adds partial caps read in from a substrate calculation in case of a interface or surface calculation.
- CONSTR_N: The N_l needed for Weinert's pseudocharge-density algorithm (see (7.14) in section 7.2.2) are chosen according to rules Weinert gives in his paper[10].
- PSEUDO: Weinert's method for finding the pseudocharge-density is executed in this routine. Upon leaving this routine the program has the plane-wave density from which the interstitial potential can be found by a simple Fourier transform.
- MULTI: The actual multipoles of the pseudocharge are calculated to check against the input.
- POTSV: The variables defined in (7.28) are calculated in preparation for solving the continuity problem.
- BCIN: The boundary condition on the external interfaces of the subvolume stack are read in.
- BULKBC/POTBC: These two subroutines perform the matching on the interstitial potential to make it continuous (see section 7.3.2). The routine BULKBC is executed for bulk systems; the routine POTBC is able to handle interface and surface systems.
- AVINT: The average interstitial potential is calculated. In a bulk system this average potential is set to zero, to make sure the energy interval used to calculate the total charge in the GROW module doesn't move too much.
- VCAP: The spherical harmonic expansion of the interstitial potential on the relevant parts of the muffin-tin spheres is calculated.
- VMTLM: The parts of spherical harmonic expansion on the muffin-tin spheres are added to form the boundary conditions for solving the radial Poisson equation.
- VMTR: The radial Poisson equation is solved.
- WORKF: In the case of a surface calculation the workfunction is calculated, by calculating the electrostatic potential on the external interface with vacuum. The assumption made here is that far away from the surface there is no charge, and therefore no exchange-correlation potential.
- XCPOT: The expansion of the exchange-correlation potential is constructed as described in section 7.4.

At this stage the potential is known. All that is left is output of certain quantities.

7.5.3 Putting the potential in a file

The output part of the FISH module is rather simple. There is the routine OUTPOT which puts the potential in a file in a format readable by both GROW and MIX. There are also several small routines which output partial multipoles, partial spherical harmonics expansions of interstitial potentials and potential components on subvolume interfaces. These outputs are sometimes needed as boundary conditions in other calculations.

7.6 Material going into the FISH pool

In this section we shall look at the different types of input going into the FISH module. Since some of the input-files of FISH are exactly the same as for GROW, I shall in those cases indicate the parameters which are really only important for the FISH module.

7.6.1 The "struct" file

The same struct file is used for both the GROW and the FISH modules. The only parameter in the struct file which is used exclusively in the FISH code is lxc, indicating which exchange-correlation algorithm to use.

7.6.2 The "rho.*" and "qlm.sv*" files

As indicated in section 6.6.3, these files contain the charge-density and multipole moments. In the case of a non-bulk calculation, there are two extra qlm.sv* files, namely one for subvolume 0, the right-most subvolume of the left substrate and one for subvolume nsv+1, the left-most subvolume of the right substrate.

7.6.3 The "leftbc.in" and "rightbc.in" files

These files contain the partial multipole moments and partial spherical harmonic expansion of the interstitial potential, in the case of a non-bulk calculation, to provide the correct match to the substrate.

7.6.4 The "potbc.left.in" and "potbc.right.in" files

In the case of a interface or surface calculation, these files contain the $V^{\infty}_{0,L}$ and $V^{\infty}_{0,R}$ needed for the boundary conditions as discussed in sections 7.3.2.2 and 7.3.2.3.

7.7 Output of the FISH module

The important output of the FISH module is of course the potential, output into the file pot.out. The boundary condition files mentioned in the previous section are also output by FISH, when it is executed in bulk mode (in that case the files have the extension .out instead of .in).

Chapter 8

MIXing

In the two previous chapters we have seen how to solve the Schrödinger equation and how to solve Poisson's equation. So in theory we can now start performing self-consistent calculations. However, in practice it is very unstable to just put output potentials into GROW *and* put output charge-densities into FISH. As was mentioned before in chapter 5, we must put a mixing step either in between GROW and FISH (mix charge-densities), or in between FISH and GROW (mix potentials). The mixing process and mixing module MIX are the subjects of this chapter.

8.1 The different roads to convergence

As we have already seen in chapter 5 (figures 5.1 and 5.2), there are two different tracks to reach convergence. The first track, which we call the *potential road*, mixes potentials (figure 5.1). The second track, which we call the *charge-density road*, mixes charge-densities (figure 5.2).

The module controlling the convergence track is MIX. It works in two distinct modes, either in potential mode or in charge-density mode. In the following section we shall look at the two modes in a bit more detail.

8.1.1 The potential roads

In the potential mode the MIX module takes two or more potentials and mixes them to form a new potential. This mixing is done by performing a mixing algorithm on the potential expansion coefficients. The current incarnation of the MIX module can actually handle two different types of potential.

8.1.1.1 The wide road of the full potential

First, there is the full potential, as has been used in this thesis up to now. This potential is able to describe almost any system - close packed or open, surfaces with adsorbates and so on. The potential in such a calculation is continuous everywhere by construction and can handle large interstitial regions.

I call the full potential a *wide road*, because it can handle almost any system, however open it is. So it can handle a *wide* variety of systems. Another way to look at this is, that all the potential coefficients constitute a wide system, with many degrees of freedom on their way to self-consistency.

Before the MIX code can do its work, it puts the entity that requires mixing into a vector which we shall call $\mathbb{X}$. This vector 'spans' the entire stack of subvolumes, i.e. all expansion coefficients of all subvolumes go into one vector. The precise order in which the elements go into the vector is not important. We just need to remember that for full potential mixing the vector $\mathbb{X}$ can becomes rather long.

One disadvantage of the full potential, as we have seen in chapter 6, is the fact that its Hamiltonian matrix elements are rather time-consuming to calculate. This disadvantage and the fact that sometimes it is unnecessary to use the full potential, have lead me to construct a version of both GROW and FISH which will work with the simplified muffin-tin potential.

8.1.1.2 The fast road of the muffin-tin potential

Before we look at the expansion of the muffin-tin potential, let us remind ourselves what the full potential looks like:

$$V^I(\mathbf{r}) = \sum_{\mathbf{g}} V^I_{\mathbf{g}} e^{i\mathbf{g}\cdot\mathbf{r}} + \sum_{\mathbf{G}} \left\{ V^{I,+}_{\mathbf{G}} e^{|\mathbf{G}|z} + V^{I,-}_{\mathbf{G}} e^{-|\mathbf{G}|z} \right\} + V^I_1 z + V^I_2 z^2 + \sum_{\alpha} \sum_{L}^{\tilde{n}_l} V^I_L(r_\alpha) Y_L(\widehat{\mathbf{r}_\alpha}) \quad (8.1)$$

where I indicates the subvolume. The first step in simplification is to assume a constant potential, V_0, in the interstitial region, so that (8.1) becomes:

$$V^I(\mathbf{r}) = V^I_0 + \sum_{\alpha} \sum_{L}^{\tilde{n}_l} V^I_L(r_\alpha) Y_L(\widehat{\mathbf{r}_\alpha}) \quad (8.2)$$

We then get rid of all non-spherical terms within the muffin-tin and are left with the muffin-tin potential:

$$V^{I,\mathrm{mt}}(\mathbf{r}) = V^I_0 + \sum_{\alpha} V^I_{\mathrm{sph}}(r_\alpha) \quad (8.3)$$

The first thing to notice is that this potential is much more limited than the full potential. The constant interstitial potential is, for example, not very suited for open systems with large interstitial regions. The spherically symmetric potential is not really suited for systems where the muffin-tins have a low coordination number, e.g. at a surface.

In the case of vacuum subvolumes, which are sometimes needed in surface or tunneling calculations, the constant interstitial potential is very bad at describing the physics. Therefore I use another form of potential for the vacuum subvolumes (empty of muffin-tins, also no muffin-tins are allowed to protrude into these subvolumes):

$$V^{I,\mathrm{vac}}(\mathbf{r}) = \sum_{g_z} V^I_{g_z} e^{ig_z z} \quad (8.4)$$

Again the MIX code constructs a vector $\mathbb{X}$ containing all the expansion coefficients in all subvolumes in the subvolume stack. This vector is a lot shorter than the one for a full potential.

Precisely because the muffin-tin potential contains fewer coefficients and also because those parts of the potential which are most expensive in matrix element determination are

not present, the entire code is a lot faster when using muffin-tin potentials. That is why I call the convergence track using the muffin-tin potential a fast (but narrow) road. The choice to be made in advance is whether we would like rather accurate results and don't mind that it takes some time to get those, or whether we want less accurate results which we can calculate rather quickly.

8.1.2 The charge density road

One of the reasons the muffin-tin code works rather well, is the fact that the charge-density is always 'full'. By this I mean that the charge-density is never stripped to a form resembling the potential in (8.3). This also means that in the adapted `FISH.MT` practically the entire full electrostatic potential is calculated and then stripped down to the form of (8.3).

Let us remind ourselves what the charge-density expansion looks like:

$$\rho^I(\mathbf{r}) = \sum_{\mathbf{g}} \rho^I_{\mathbf{g}} e^{i\mathbf{g}\cdot\mathbf{r}} + \sum_{\alpha} \sum_{L}^{n_l} \rho^I_{L,\alpha}(r_\alpha) Y_L(\widehat{\mathbf{r}_\alpha}) \tag{8.5}$$

where I again indicates the subvolume. The `MIX` module puts the coefficients of all the different expansions of all the subvolumes in the stack into the vector $\mathbb{X}$, just as for the potential. This means that once these vectors are filled with either a potential (full or muffin-tin) or a charge-density the `MIX` module can use whatever mixing algorithm it has available on the vectors representing the different inputs without having to distinguish between them.

In the case of a charge-density though, we do not just need the coefficients of (8.5) to be put into $\mathbb{X}$. We also need to put the multipoles belonging to the charge-densities $\rho^I(\mathbf{r})$ into the vector, since those will be needed as inputs to the `FISH` module.

8.2 The different mixing modes

Once the vectors $\mathbb{X}_i$ for the different inputs are filled, the `MIX` module can start mixing. At the moment only two different mixing modes have been implemented in the `MIX` module: simple mixing and Broyden mixing. Both will now be discussed in more detail.

8.2.1 Simple mixing

In simple mixing, the old vector (i.e. the potential or charge-density which was used as input for this iteration) is stored in $\mathbb{X}_1$. The new vector which is the output of this iteration is stored in $\mathbb{X}_2$. The module then constructs the vector $\mathbb{F}$:

$$\mathbb{F} = \mathbb{X}_2 - \mathbb{X}_1 \tag{8.6}$$

The mixing process then becomes very simple, because it boils down to the addition of vectors:

$$\mathbb{X}_{\text{out}} = \mathbb{X}_1 + \beta\mathbb{F} \tag{8.7}$$

where β is the mixing parameter. This mixing parameter can vary between 0 and 1, and is normally quoted as a percentage.

Simple mixing is a very stable process if β is sufficiently small. However, small β means that the method can take a rather long time to reach convergence. Other techniques have been developed, to speed up the convergence, and we now turn to the method developed by Broyden[28].

8.2.2 Broyden mixing

Let us first examine what we actually want to solve when we are trying to reach convergence. When the system is converged, we have:

$$\mathbb{F} = 0 \tag{8.8}$$

So we are in essence looking for the root of a large system of equations. A standard way to approach this problem is the Newton-Raphson method, and this is the basis of the Broyden method. If we call the $\mathbb{F}$ vector at the end of the n^{th} iteration $\mathbb{F}^{(n)}$, then the starting point is to approximate $\mathbb{F}^{(n)}$ according to:

$$\mathbb{F}^{(n)} = \mathbb{F}^{(n-1)} + \mathbb{J}^{(n-1)} \left(\mathbb{X}^{(n)} - \mathbb{X}^{(n-1)} \right) \tag{8.9}$$

where $\mathbb{J}^{(n-1)}$ is the Jacobian matrix evaluated at $\mathbb{X}^{(n-1)}$. If we now assume that in the next $(n+1)^{\text{th}}$ iteration convergence is reached, then the input to the $(n+1)^{\text{th}}$ iteration, or the output of the mixer in the n^{th} iteration, should be:

$$\mathbb{X}_{\text{out}}^{(n)} = \mathbb{X}^{(n)} + \mathbb{G}^{(n)} \mathbb{F}^{(n)} \tag{8.10}$$

With $\mathbb{G}^{(n)}$ equal to the inverse of $-\mathbb{J}^{(n-1)}$, this is the well known Newton-Raphson expression. If we put $\mathbb{G}^{(n)}$ equal to β we get the simple mixing expression.

The method developed by Broyden[28] avoids explicit calculation of the Jacobian matrix. Instead an estimate is updated at each iteration without extra calculation or function evaluation. I shall not give its derivation, which can be found in the original Broyden paper and in the paper by Srivastava[29], who has improved upon the original method. The method implemented in the `MIX` module is Srivastava's adaptation of Broyden's second method as implemented by Van Leuken[30]. This method needs to store several vectors containing information about previous iterations.

In theory the Broyden mixing scheme could use all previous iterations to calculate the next input $\mathbb{X}$, but in practice it works better if the procedure is restarted every 5 or 10 iterations. Restarting means that we forget all about the previous iterations and start anew. When the Broyden procedure starts, the first mix is always a simple mix, with a given β.

8.3 The structure of the `MIX` module

The structure of the `MIX` module is fairly simple. There is a clear input part where the structure and input potentials or charge-densities are read, a calculation part which performs the mixing procedure we request and an output part which writes out the output potential or charge-density.

8.3.1 The input and initialisation part

The input part consists of three subroutines, which are:

- `PARIN`: This routine reads the control file `param.mix` which specifies the type of quantity we want to mix, what method to use, etc. Let us look at some of the control parameters in more detail:

- `mixmode` This variable can take two values
  - 0 - mix potentials
  - 1 - mix charge-densities
- `sizemode` This variable controls the action of the `MIX` module in the case of 2 inputs of different size expansions The different values are
  - 0 - mix only equal size potentials or charge-densities
  - 1 - mix to smallest size
  - 2 - mix to biggest size
- `mixing` The mixing method is controlled by this parameter, which has two options at present
  - 0 - use simple mixing
  - 1 - use Broyden's method for mixing
- `mixpar` This is the β for the simple mixing process

- `INPUTSTR` This subroutine reads the structure of the system Important variables in this case are the number of subvolumes in the stack, the number of atoms, the definition of the parallel basis, etc
- `INPUTS` This subroutine calls routines which input the potentials (`INPOT`) or charge-densities (`INCHDN`) depending on the value of `mixing`

8.3.2 The mix part

The actual mix part of the code is controlled from the routine `DOMIX`, which calls either `MIXPOT` or `MIXCHDN` These routines then each call routines for the different mixing methods As an example we shall examine the structure of the mix part of `MIX` for Broyden mixing of potentials

When we are using Broyden's method to mix potentials, the code runs through the following subroutines

- `BROYD_PARAM` One single variable is read by this routine, the number of previous Broyden iterations stored on file
- `GENVEC_POT` This subroutine constructs the vectors $\mathbb{X}_1$ and $\mathbb{F}$, using the expansion coefficients of both potentials that have been input
- `BROYD` This is the actual Broyden mixing routine It reads the vectors of previous Broyden iterations from file, writes some new vectors to file and calculates $\mathbb{X}_{out}$
- `UNPVEC_POT` The vector $\mathbb{X}_{out}$ is now unpacked into the expansion coefficients of the output potential
- `CONV_POT` This routine checks whether the potential is converged, i e whether the following relations holds

$$\sqrt{\frac{\mathbb{F}\ \mathbb{F}}{\mathbb{X}_1\ \mathbb{X}_1}} \leq \delta \tag{8 11}$$

This means that the relative change in the potential has gone below some requested tolerance

8.3.3 The output part

The output part of MIX is one subroutine, the routine OUTPUT, which calls either OUTPOT of OUTCHDN, depending on the value of mixing

8.4 The basic materials going into the bowl

The materials going into the MIX module consist of one part which goes into the module in both modes, and a part which is different for each mode

8.4.1 Standard ingredients

The standard ingredients are

- the struct file, is the same file as for the GROW and FISH module
- the param mix file This file contains the parameters read in by the routine PARIN
- the broyd param file This contains the number of previous Broyden iterations as read by BROYD_PARAM
- the file broyd tmp1, broyd tmp2 and broyd tmp3 These files contain the previous vectors saved and needed by the Broyden method

8.4.2 Ingredients for potential mixing

When we are cooking up a potential, we also need some potential inputs The potentials are read in from files, whose names are stored in the param mix file The first potential read in is stored in $\mathbb{X}_1$ and is assumed to be the input potential to the present iteration The second potential is stored in $\mathbb{X}_2$ and is assumed to be the output potential of FISH is the current iteration

8.4.3 Ingredients for charge-density mixing

For cooking charge-densities, we need to input charge-densities The names of the files are again read in from param mix, and the order of the charge-densities is the same as the order of the potentials as described in the previous section However, when mixing charge-densities, we also need to mix the corresponding multipole moments Again the names of the files containing the multipole moments are read from param mix in the same order as for the charge-densities

8.5 The products in the bowl after mixing

Once the mixer has stopped, we are left with the mixed products These products are either a potential, or a charge-density and corresponding multipole moments The potential is written to the file pot out The charge-density is written to rho int and rho mt, and the multipole moments are stored in qlm sv?

Chapter 9

How to count electrons using the GROW module

In this chapter we shall discuss some of the modifications and extensions of the GROW module needed to be able to calculate conductances, as described in chapter 4.

For bulk calculations the most important change is a very precise determination of the real bands and the calculation of the transformation matrix $\mathbb{U}$ between the intersubvolume basis and the Bloch basis.

For non-bulk system the adapted GROW code has to calculate the transmission probability matrix for the Bloch states to the left of the subvolume stack to the Bloch states on the right of the subvolume stack.

In this chapter I shall call the adapted GROW module: GROW.COND.

9.1 Setting up the Bloch electrons from the bulk

As we have seen in sections 4.2.1 and 6.4.2.1, the GROW module can calculate a band-structure. In section 6.4.2.1 we mentioned that the real band-structure is selected by taking all those k_z values that have an imaginary part smaller than a certain tolerance. The choice of tolerance can occasionally result in eigenvalues being missed. When we only want to look at a band-structure to get a feeling for a material, that procedure is fine, since a point missing here or there doesn't change the overall picture. However, when we are interested in the travelling Bloch states in a system, we really need to find *all* the real states, no more and no less.

So before we can even start to do anything about calculating conductances, we need to find an algorithm, which will always give us the real bands at a given energy. First we examine again what happens in the CALC_BANDS routine of the GROW code.

9.1.1 How we constructed the band-structure in GROW

In the GROW module, the transfer matrix for the entire subvolume stack is constructed for a given energy $\mathcal{E}$. For this transfer matrix we have the property:

$$\mathbb{T} b_{\mathbf{k}}(\mathbf{r}) = e^{i\mathbf{k}\,\mathbf{r}_t} b_{\mathbf{k}}(\mathbf{r}) \tag{9.1}$$

So the Bloch states are the eigenvectors of the transfer matrix, and the eigenvalues of the transfer matrix give us the band-structure. The k_z values we get from the eigenvalues of $\mathbb{T}$

are not all real - actually most are complex. These complex k_z belong to what are called evanescent states, states decaying or increasing exponentially with z. In a bulk crystal these states cannot propagate, so we are only interested in those Bloch states that correspond to 'real' k_z.

The problem we face is that since we are using a finite imaginary energy, the k_z are never completely real. Therefore we need to devise a method to find those states which would normally have zero imaginary part. In the routine CALC_BANDS in the GROW module, this is done by picking the k_z which have an imaginary part below a certain tolerance and treat these as real. The k_z chosen in this way then form the band-structure.

This band-structure can sometimes have gaps, because for a certain energy the 'real' bands have a imaginary k_z which just jumps above the tolerance. Setting the tolerance higher means that we sometimes include complex bands into the band-structure. As we have already mentioned, for conductance calculations we need *all* the real k_z and *only* the real k_z, so this algorithm is unsatisfactory. The new algorithm which has been devised is described in the next section.

9.1.2 Using imaginary energy variation to find the real bands

The idea at the basis of the new algorithm is the fact that the normally real k_z has a small imaginary part because of an imaginary part of the energy. For 'real' bands the imaginary part of k_z is proportional to the imaginary part of the energy. We can prove this by a Taylor expansion argument. We expand the k_z as a function of imaginary energy around the real part of the energy:

$$k_z(E + \imath\epsilon) = k_z(E) + \imath\epsilon\frac{\mathrm{d}k_z}{\mathrm{d}E} + \mathcal{O}(\epsilon^2) \tag{9.2}$$

We see that the imaginary part of k_z indeed varies linearly with the imaginary part of the energy. For the complex bands, the $k_z(E)$ is already complex, so in that case the term linear in ϵ is a higher order effect. Let us examine the new algorithm for finding the real bands in a bit more detail:

1. We have a certain energy $\mathcal{E} = E + \imath\epsilon$.
2. We now increase the imaginary part of the energy by a factor f, so it becomes $\mathcal{E}' = E + \imath f\epsilon$
3. We solve for the Green functions for $\mathcal{E}'$, and construct the scattering matrices.
4. Then we construct the transfer matrices for the subvolumes, and multiply those to get the transfer matrix $\mathbb{T}(\mathcal{E}')$ for the entire subvolume stack.
5. We can diagonalise this transfer matrix, to give us a set of bands, $\{k_z(\mathcal{E}')\}$.
6. We go back to the original energy $\mathcal{E} = E + \imath\epsilon$.
7. We go through the entire process in steps 3 and 4, this time for $\mathcal{E}$.
8. Diagonalising $\mathbb{T}(\mathcal{E})$ gives us a set of bands, $\{k_z(\mathcal{E})\}$.
9. We have to make sure that the two sets of bands have the same ordering.

10. We can now compare these sets of bands. Those k_z for which the imaginary part has varied linearly with the variation from $\mathcal{E}'$ to $\mathcal{E}$, are the real bands.

This process works, but ..... it is rather expensive. We have to solve for the Green function for two different energies, which means we have two big inversions for every subvolume for every real energy and every $\mathbf{k}_{||}$. However, at the moment that is the price we have to pay. No other cheaper solutions to this problem have been found yet.

Once we have found the real k_z, we know which of the Bloch states are travelling states. These Bloch states are the ones that can carry current. In setting up the transformation matrix $\mathbb{U}$ we sort the Bloch states into those moving to the right and those moving to the left. Within those two groups of Bloch states, the travelling waves are sorted at the top, the evanescent waves are ordered with respect to their imaginary part of k_z.

9.2 Counting Bloch electrons

For systems with bulk periodicity we can now perform the following calculation:

- Set the conductance, $G = 0$.
- Choose a set of $\mathbf{k}_{||}$ which covers the irreducible part of the Brillouin zone.
- For every $\mathbf{k}_{||}$ set the energy to the Fermi energy.
- Use the process described in the previous section to find the Bloch states and matrix $\mathbb{U}$.
- Write $\mathbb{U}$ to file, for use in later calculations.
- Count the number of Bloch states travelling towards the right for every $\mathbf{k}_{||}$, we shall call this $N_{\mathbf{k}_{||}}$.
- After we have done this for all $\mathbf{k}_{||}$, perform the following sum:

$$\frac{G}{S} = \frac{2e^2}{Ah} \sum_{\mathbf{k}_{||}} N_{\mathbf{k}_{||}} \tag{9.3}$$

We have now calculated the conductance per unit area for the bulk system.

We can calculate the conductance for bulk periodic systems like this, because every travelling Bloch state can hold one electron (per spin). So counting travelling Bloch states tells us how many electrons can conduct current in a given direction.

This is nothing new. Counting these states can be done by any method which can give an accurate band-structure at the Fermi energy. However, what happens when the system we are interested in no longer has bulk periodicity? For those systems we have to do more than just counting.

9.3 Conductance is more than just counting

As we have seen in chapter 4, the Landauer conductance formula can be written as:

$$\frac{G}{S} = \frac{2e^2}{Ah} \sum_{\mathbf{k}_{||},\nu\mu} |t_{\mathbf{k}_{||}\mathbf{k}_{||},\nu\mu}|^2 \tag{9.4}$$

where S is the area through which we conduct and A is the area of the two-dimensional unit-cell. The matrix $t_{\mathbf{k}_\parallel \mathbf{k}_\parallel,\nu\mu}$ is the transmission probability from a state μ with wavevector $\mathbf{k}_\parallel$ on the left to a state ν with wavevector $\mathbf{k}_\parallel$ on the right. In the previous section in the system with bulk periodicity, the matrix $t_{\mathbf{k}_\parallel \mathbf{k}_\parallel,\nu\mu}$ was diagonal and equal to the unit matrix, thus we could just count the number of states. In most interesting systems, however, we do not have this three dimensional periodicity. In these systems we shall have to calculate $t_{\mathbf{k}_\parallel \mathbf{k}_\parallel,\nu\mu}$.

In section 4.4 I already sketched an outline of how to find the matrix $t_{\mathbf{k}_\parallel \mathbf{k}_\parallel,\nu\mu}$ for an arbitrary interface system. I shall now describe the changes which have been made to the GROW module to be able to calculate the conductance, using (9.4).

9.3.1 The new CALC_BANDS routine

The *new* CALC_BANDS subroutine uses the algorithm defined in section 9.1.2 to find the real bands. This routine is only called in bulk-system-mode, i.e. for the substrates of the interface in question. For every $\mathbf{k}_\parallel$ it writes the following entries to the new file umat.left:

1. The $\mathbf{k}_\parallel$ vector, for checking purposes when this file is input in the interface calculation.
2. The number of parallel wavevectors in the intersubvolume basis, nrec. So there are 2 * nrec Bloch states, nrec moving to the left and nrec moving to the right.
3. The matrix $\mathbb{U}$, which is a (2 * nrec) $\times$ (2 * nrec) matrix.
4. The number of conducting Bloch states travelling from left to right.
5. For each of those conducting Bloch states the index in the $\mathbb{U}$ matrix.
6. Also for each of the conducting Bloch states its flux, calculated using (4.27).
7. The number of conducting Bloch states travelling from right to left. For non-symmetric bulk systems this is not the same as point 4.
8. The index of those states in the matrix $\mathbb{U}$.
9. The flux in those states.

This is the only change to GROW which is needed when running in a bulk configuration. The actual work for the interface is done in an entirely new subroutine CONSTR_RT_COND2, called from DOENERGY, which is described in the next section. Actually this new routine is also called in the bulk mode, but is primarily used for testing purposes.

9.3.2 The new routine CONSTR_RT_COND2

This routine does the actual work. The left-to-right transfer matrix is already calculated in GROW, in subroutine CONSTR_RT. Thus we have to put our new subroutine in the new module GROW.COND, somewhere after the call to CONSTR_RT.

Before we can call this new routine, the DOENERGY subroutine reads both the umat.left and umat.right files. The file umat.left contains the information for the left substrate and the file umat.right the information for the right substrate.

This means we now have the following information:

- The transformation matrix between the intersubvolume basis and the Bloch states in the left substrate, $\mathbb{U}_L$.
- The transformation matrix between the intersubvolume basis and the Bloch states in the right substrate, $\mathbb{U}_R$.
- The number of conducting Bloch states on a certain side of the subvolume stack travelling in a certain direction. These numbers have names like `nrchannel_??`, where the last two letters indicate on which side the states are (left or right) and in which direction the states travel, e.g. states on the left travelling to the right would have the suffix `lr`.
- The fluxes in all the conducting Bloch states, with names with an ending similar to the above, e.g. `vnormlr()` is the array containing the fluxes of the Bloch states on the left of the subvolume stack travelling to the right.

Using all this information, and the information that was already present in the old `GROW`, the module `GROW.COND` can then call `CONSTR_RT_COND2` to calculate the transmission probability matrix $t_{\mathbf{k}_{\|}\mathbf{k}_{\|},\nu\mu}$.

The first step is the calculation of the left-to-right transfer matrix in terms of Bloch states to the left and right of the subvolume stack. This is done using expression (4.19), or in our current notation:

$$\mathbb{T}^{\rightarrow}_{\text{Bloch}} = \mathbb{U}_R^{-1}\mathbb{T}^{\rightarrow}\mathbb{U}_L \tag{9.5}$$

where $\mathbb{T}^{\rightarrow}$ is the left-to-right transfer matrix in terms of intersubvolume basis functions. From (3.62) we know that the left-to-right transfer matrix contains the reflection and transmission matrices:

$$\mathbb{T}^{\rightarrow}_{\text{Bloch}} = \begin{pmatrix} \mathbb{T}^{\rightarrow}_{11} & \mathbb{T}^{\rightarrow}_{12} \\ \mathbb{T}^{\rightarrow}_{21} & \mathbb{T}^{\rightarrow}_{22} \end{pmatrix} = \begin{pmatrix} \left(T^L - R^R(T^R)^{-1}R^L\right) & R^R(T^R)^{-1} \\ -(T^R)^{-1}R^L & (T^R)^{-1} \end{pmatrix} \tag{9.6}$$

where the scattering matrices are now in terms of Bloch states.
To find the T^R matrix, we use:

$$T^R = (\mathbb{T}^{\rightarrow}_{22})^{-1} \tag{9.7}$$

Using this T^R, we obtain R^L using:

$$R^L = -T^R\mathbb{T}^{\rightarrow}_{21} \tag{9.8}$$

Similarly for R^R:

$$R^R = \mathbb{T}^{\rightarrow}_{12}T^R \tag{9.9}$$

And finally for T^L:

$$T^L = \mathbb{T}^{\rightarrow}_{11} - R^R\mathbb{T}^{\rightarrow}_{21} \tag{9.10}$$

When the transmission is rather small, as for example in tunneling calculations, it is more stable to use an equivalent definition for the (11) block of $\mathbb{T}^{\rightarrow}$, which is:

$$\mathbb{T}^{\rightarrow}_{11} = \left(T^{L\dagger}\right)^{-1} \tag{9.11}$$

so that we can use:

$$T^{L} = \left[(\mathbb{T}^{\rightarrow}_{11})^{-1}\right]^{\dagger} \tag{9.12}$$

The scattering matrix T^L is the one we want. This matrix gives the transmission of a Bloch state on the left to a Bloch state on the right of the subvolume stack. Before we can use it, it has to be normalised. Since the Bloch states do not carry unit flux, we have to renormalise the transmission matrix. If I^{rr}_{ν} is the flux in state ν to the right of the subvolume stack travelling to the right, and I^{lr}_{μ} is the flux in state μ on the left of the stack, travelling to the right, then we renormalise using (see (4.28)):

$$t_{\mathbf{k}_{\|}\mathbf{k}_{\|},\nu\mu} = \left[T^{L}\right]_{\nu\mu} \sqrt{\frac{I^{\mathrm{rr}}_{\nu}}{I^{\mathrm{lr}}_{\mu}}} \tag{9.13}$$

We now have the elements of the big t matrix. The routine CONSTR_RT_COND2 writes out the elements $t_{\mathbf{k}_{\|}\mathbf{k}_{\|},\nu\mu}$ to the file transmat. This file can then be read by a separate program to generate the conductance using (9.4).

So we now have given a description of the implementation of the conductance theory of chapter 4 in a new version of the GROW module, called GROW.COND. This new module runs stand-alone, i.e. not in a self-consistency loop. The usual process for calculating the conductance in an interface is to execute GROW.COND for the two substrates, and use the output of those runs as input for a run of GROW.COND on the interface configuration.

The new GROW.COND module can handle all geometries which the GROW module can handle, and there are both a full-potential and a muffin-tin potential version. However, we would like to perform calculations (both self-consistent and conductance) for magnetic systems. So the next step, described in the following chapter, is the implementation of a spin-polarised version of the entire code.

Chapter 10

How to implement a spin-polarised version

In this chapter we shall examine how to implement a spin-polarised version of the method described in part I. A spin-polarised version of the code is necessary when we want to examine magnetic properties of materials. Before we implement this, we shall look at spin in general.

10.1 Spin in quantum mechanics

To be able to handle spin, we introduce the spinor, well-known from quantum mechanics.

10.1.1 What is a spinor, and how do we use it?

In general, spin is introduced as an operator **S**. The eigenstates of the z component of this operator S_z are represented by a two component column vector, called a *spinor*[31, 13], i.e.

$$S_z\chi = \pm\tfrac{1}{2}\hbar\chi \tag{10.1}$$

where χ is a spinor, which can be written as:

$$\chi = \begin{pmatrix} u \\ v \end{pmatrix} \tag{10.2}$$

With the most often used matrix representation of the S_z operator, we obtain:

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = \pm\begin{pmatrix} u \\ v \end{pmatrix} \tag{10.3}$$

From this last equation we can find the eigensolutions:

$$\chi_+ = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \chi_- = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{10.4}$$

for the eigenspinors corresponding to spin-up ($S_z = +\frac{1}{2}\hbar$) and spin-down ($S_z = -\frac{1}{2}\hbar$), respectively. An arbitrary spinor can now be represented in this basis:

$$\begin{pmatrix} \alpha_+ \\ \alpha_- \end{pmatrix} = \alpha_+\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \alpha_-\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \alpha_+\chi_+ + \alpha_-\chi_- \tag{10.5}$$

10.1.2 How does our basis change?

Once we have introduced spin, we see that all wavefunctions in general have spinor form[31]:

$$\left|\Psi\right\rangle = \begin{pmatrix}\psi_+(\mathbf{r})\\ \psi_-(\mathbf{r})\end{pmatrix} \tag{10.6}$$

We notice that if we want to represent these wavefunctions in terms of a basis $\left|\Phi_\mathbf{g}\right\rangle$, we have to use a new definition of the basis:

$$\left|\Phi_\mathbf{g}\right\rangle = \begin{pmatrix}\phi_{\mathbf{g},+}(\mathbf{r})\\ \phi_{\mathbf{g},-}(\mathbf{r})\end{pmatrix} = \phi_\mathbf{g}(\mathbf{r})\chi_+ + \phi_\mathbf{g}(\mathbf{r})\chi_- \tag{10.7}$$

where $\phi_\mathbf{g}(\mathbf{r})$ are the basis functions we have already used for the non-spin-polarised method. From this we see that the basis essentially doubles in size.

10.1.3 What are the consequences for the Hamiltonian?

The doubling of the basis has an effect on the Hamiltonian matrix. Without looking at the exact form of H, let us examine its matrix elements defined as:

$$H_{(\mathbf{g},s);(\mathbf{g}',s')} = \int \mathrm{d}\mathbf{r}\chi_s^*\phi_\mathbf{g}^*(\mathbf{r})H(\mathbf{r})\phi_{\mathbf{g}'}(\mathbf{r})\chi_{s'} \tag{10.8}$$

From this we can see that the Hamiltonian matrix becomes four times as big (twice in both directions) as for a non-spin-polarised description. In terms of the basis described in (10.7), we can write it as:

$$H_{(\mathbf{g},s);(\mathbf{g}',s')} = \begin{pmatrix}H_{uu} & H_{ud}\\ H_{du} & H_{dd}\end{pmatrix} \tag{10.9}$$

In (10.9) we list the basis functions in such a way that we first have all the *up* basis functions and then all the *down* basis functions.

The Hamiltonian for a spin-polarised system (without any external fields) has the form[14]:

$$H = -\tfrac{1}{2}\nabla^2 + \begin{pmatrix}V_\mathrm{up}(\mathbf{r})\\ V_\mathrm{down}(\mathbf{r})\end{pmatrix} + \alpha\mathbf{L}\cdot\mathbf{S} \tag{10.10}$$

where the second term is the potential (the split in up and down is due to exchange) and the third term is due to spin-orbit coupling. Since the spin-up and spin-down spinor are perpendicular, i.e.

$$\left\langle\chi_+\middle|\chi_-\right\rangle = \left\langle\chi_-\middle|\chi_+\right\rangle = 0, \tag{10.11}$$

the first term on the right-hand-side of (10.10) contributes identically only to the H_{uu} and H_{dd} blocks of the matrix in (10.9). The second term, the spin-dependent potential also contributes only to H_{uu} and H_{dd}. the only term contributing to the off-diagonal element H_{ud} and H_{du} is in fact the spin-orbit term. However, in our method we do *not* include spin-orbit interaction, since for the materials and properties we are interested in, the effects of the spin-orbit interaction are largely negligible. This means that the matrix in (10.9) is block-diagonal.

10.1.4 The spin-polarised embedding potential

By analogy with the potential in (10.10) and from the argument that the Green function has the same block-diagonal form as the Hamiltonian, we can see that we have two embedding potentials, spin-up and spin-down. Using the same argument, all the scattering matrices will have a spin-up and a spin-down form as well.

10.1.5 Local spin-density approximation

The potential in (10.10) is spin-dependent. The electrostatic potential, which depends on all charge independent of spin, cannot be spin-dependent, the spin-dependence comes from the exchange-correlation potential.

It is known from density functional theory, that when spin is included, the energy becomes a functional of the spin density matrix. In the local approximation that we use, the local density approximation becomes the local spin-density approximation[25, 24](LSDA). In the LSDA, the algorithms take as input variables the spin-up and spin-down charge-density at a point in space, and return the spin-up and spin-down exchange-correlation potential (and energy density) at that point.

The methods for calculating exchange-correlation potentials which we have implemented in routine `EXCORR` in module `FISH` (see section 7.4) can all be adapted from LDA to LSDA. In fact the routines incorporated in `EXCORR` were originally written for a spin-polarised code, and up to now have been used for a single spin.

10.2 Can we include spin with as little work as possible?

We have seen that the new spin-polarised Hamiltonian matrix is block diagonal, with one spin-up and one spin-down block. To implement spin with as little changes to the existing code as possible we treat each spin separately. Each spin has a normal size Hamiltonian depending on the potential and embedding potentials for that spin. Each Hamiltonian leads to the Green function for that spin direction, which leads to the charge density for that spin direction. Only at the stage of self-consistency do we need to bring both spin-directions together.

So let us now write down how to implement a spin-polarised version of each of the modules of the code.

10.2.1 Spin polarising the `GROW` module

In adapting the `GROW` module to the `GROW.MAG` module, we use the following tricks:

- Once the $\mathbf{k}_\parallel$ set has been read in/built, we double the set. So if we were normally to have N $\mathbf{k}_\parallel$-points: $\{\mathbf{k}_\parallel{}^1, \mathbf{k}_\parallel{}^2, \ldots, \mathbf{k}_\parallel{}^{N-1}, \mathbf{k}_\parallel{}^N\}$, we now have the set of $2N$ $\mathbf{k}_\parallel$-points: $\{\mathbf{k}_\parallel{}^1, \mathbf{k}_\parallel{}^2, \ldots, \mathbf{k}_\parallel{}^N, \mathbf{k}_\parallel{}^1, \mathbf{k}_\parallel{}^2, \ldots, \mathbf{k}_\parallel{}^N\}$.
- The first half is the spin-up half, the second the spin-down.
- Use the first N $\mathbf{k}_\parallel$-points with the spin-up potential and embedding potential as we would in the non-spin-polarised module. We have to take care that all the factors of 2, which were originally implemented to take care of two electrons per state, are removed.

- Once we have finished the N $\mathbf{k}_{\|}$-points, we can now construct quantities like the spin-up charge-density, the spin-up density of states, etc.
- We reset all parameters and variables which are not linked to total quantities, i.e. quantities that are summed for spin-up and spin-down.
- We now input the spin-down potential and embedding potentials.
- Run through the second N $\mathbf{k}_{\|}$-points and calculate the Green functions and densities of states, etc.
- We now have all the spin-up and spin-down quantities, as well as the spin-summed quantities.
- We can write everything to files: all output quantities which the `GROW` module gave are now given for both spin-up and spin-down separately.

By implementing spin-polarisation in this way, the code has to do almost exactly twice as much work. It is also possible to make all relevant changes to `GROW` and implement a spin-polarised version in very little time and with few extra problems.

10.2.2 `FISH` handles spins as well

The changes in the `FISH` module are even smaller. As we have mentioned before, the electrostatic potential is spin-independent, depending only on the total charge-density. The only part of the `FISH` module which changes upon introduction of spin is the exchange-correlation potential part.

If we examine section 7.4 again, we see that we can easily solve for both exchange-correlation potentials at once. We need the spin-up and spin-down charge-density at all random points, but these can easily be constructed from the output of the `GROW.MAG` module.

Once we have both exchange-correlation potentials, we can add them to the electrostatic potential and we have the two spin-dependent potentials. These then become the most important output of `FISH.MAG`.

10.2.3 How `MIX` should handle spin

The changes to the mixing module `MIX` are almost trivial. This module literally has to do its work twice. Whether it is mixing potentials or charge-densities, it has to do the mixing for both the spin-up and spin-down.

Actually, in the case of Broyden mixing, both the spin-up and spin-down entities (either potentials or charge-densities) are stored in one vector $\mathbb{X}$, but other than that there are no fundamental difficulties with the `MIX.MAG` module.

10.3 The spin-polarised code has some new ins and outs

Since the spin-polarised code outputs some quantities in both spin directions, some of the files have changed names. In the following two sections I shall list the old files and indicate their new names.

10.3.1 The new or renamed input files

I shall list the old files, grouped by the original module.

- The files that were input to `GROW` remain input to `GROW.MAG`, except for:
  - pot: This file becomes the two input files pot.up and pot.down.
  - rmat*: These files retain the same name, but now contain both up and down scattering matrices, ordered by the doubled $\mathbf{k}_{\|}$-set.
  - embpot*: Same as for the rmat*-files.
- The input to `FISH.MAG` is the same as for `FISH`, except for:
  - rho.*: Apart from the total charge-density, which is still stored in the same files, new files are added for the separate spin charge-densities. These files get the suffix .up and .down.
- All input files to `MIX.MAG` remain the same, apart from the potentials or charge-densities, which are doubled and get the suffices as described for `GROW.MAG` and `FISH.MAG`.

10.3.2 The new or renamed output files

The output files are changed as follows:

- The output files of `GROW` which have changed and have acquired either a .up or .down suffix are:
  - bands.*
  - rho.*
  - dos.sv?.*
- The only output of `FISH` which is affected is the potential, which is output as pot.up and pot.down.
- `MIX.MAG` outputs spin-split potentials or charge-densities as mentioned before.

10.4 Do we need so many versions of all modules?

It now seems that we have several versions of all the modules. We have already mentioned some, and others were implied:

1. `GROW`, `GROW.MT`, `GROW.MAG`, `GROW.MAG.MT`, `GROW.COND`, `GROW.COND.MT`
2. `FISH`, `FISH.MT`, `FISH.MAG`, `FISH.MAG.MT`
3. `MIX`, `MIX.MT`, `MIX.MAG`, `MIX.MAG.MT`

This is a total of 14 modules. However, once we realise that the magnetic versions can also handle non-spin-polarised systems, controlled by one single input parameter, the number of different modules gets a lot smaller. And also the full-potential (no extension) versions share most of their source code with the muffin-tin potential versions (`.MT` extension). So the modules which are left and will be used in *Part III: Results* are (grouping together as used):

1. GROW.MAG , FISH.MAG , MIX.MAG
2. GROW.MAG.MT, FISH.MAG.MT, MIX.MAG.MT
3. GROW.COND, GROW.COND.MT

Part III

Results

Chapter 11

The 'simple' systems....which took the longest

In this chapter I shall describe the first calculations done using the new code. These calculations were performed as test calculations and thus took quite long to get right, because during these tests many implementation errors reared their ugly heads.

11.1 The Al(100) test surface

The first test system that we choose is the Al(100) surface, typical of a nearly-free-electron system. Before we can start the surface calculation, we need to do a bulk calculation. We shall examine this first step in the following section.

11.1.1 First bulk Al

Aluminium has the face-centered-cubic (fcc) crystal structure, and we divide the solid into layers perpendicular to the (100) direction. This means that the atoms within the layers are ordered as in figure 11.1. From this figure it is clear that the two-dimensional (2D) lattice is square, with one atom per unit cell. The next layer is shifted by $(\frac{1}{2}, \frac{1}{2})$ in terms of the basis of the 2D lattice. We can now define the structure of the aluminium bulk calculation, by specifying the layers as shown in figure 11.1 and the layer-to-layer vector of $(\frac{1}{2}, \frac{1}{2}, z_n)$. We then first converge the bulk system using a muffin-tin potential. Using the Broyden mixing method with a base mixing of 2%, this reaches convergence (to within 10^{-5}) in about 40 iterations. This gives us a converged bulk charge-density, which we can use as a starting point for the full potential calculation. In this example and in all examples mentioned later on, we iterate by mixing potentials. Once we converge that (again taking about 30 to 40 iterations) we have the aluminium bulk system. We can then calculate all sorts of properties for that system, for example the density of states in the embedding region for $\mathbf{k}_{\|} = (0, 0)$, as shown in figure 11.2. We see in this figure, that the density of states is a smooth function. This is not because of smearing out the DOS, but a direct result of the embedding method, which gives us real bulk continua. In these calculations, and in fact in all calculations in this chapter, the Hedin-Lundqvist form of LDA is used. Now that we have the converged full potential for the aluminium bulk, we can turn to the surface.

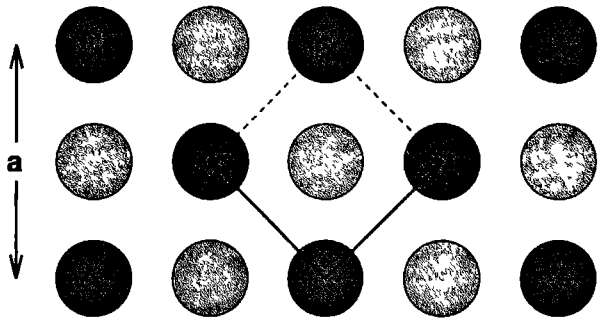


Figure 11.1: *Structure of atoms within the Al(100) layers. The dark shaded circles indicate the atoms in this layer, lighter shaded circles are the atoms in the layers above and below this layer. The dashed lines (together with the arrows) show the unit cell. The basis vectors which we choose for the two-dimensional lattice are depicted by the arrows.* **a** *is the cubic lattice constant.*

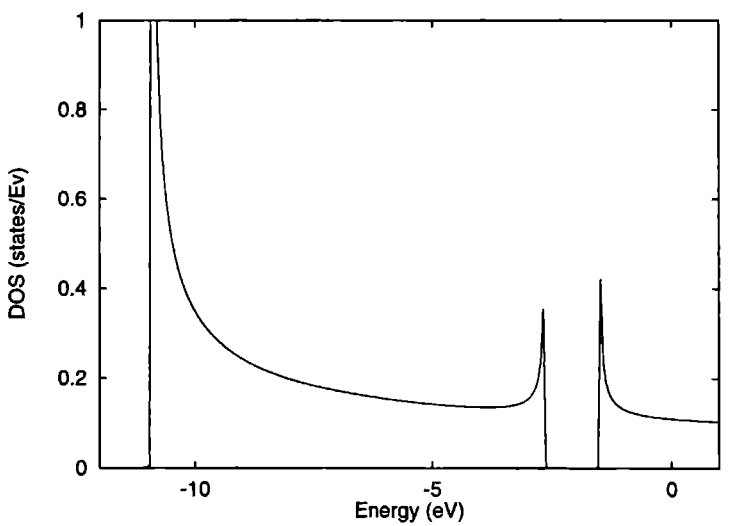


Figure 11.2: *Density of states for the* $\bar{\Gamma}$ *point of the 2D Brillouin zone,* $\mathbf{k}_{\|} = (0,0)$. *The imaginary energy for this DOS is* 1.0×10^{-5} *Hartree. The zero of energy is the Fermi energy.*

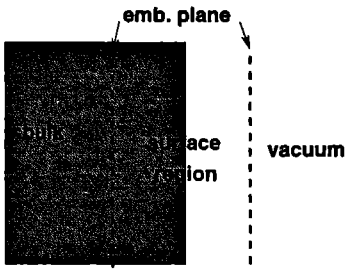


Figure 11.3: *Sketch of the surface calculation. The embedding potentials replace the bulk and vacuum to left and right of the surface region. In the surface region we can build up the system from multiple subvolumes.*

11.1.2 Then we construct the Al(100) surface

For the bulk aluminium in the previous section we used bulk periodic boundary conditions. For the surface we use the boundary conditions from the bulk calculation, i.e. embedding potentials, reflection matrices and partial multipoles and potential values on the embedding plane, to couple the surface region to the bulk. On the vacuum side we have an embedding potential and reflection matrix (calculated analytically, see (3.116)) which are updated in each iteration for the proper vacuum level. In general the situation for a surface calculation is as shown in figure 11.3. We can choose as many subvolumes within the surface region as we want.

11.1.2.1 Using 2 sub-volumes

The smallest surface region we can use consists of one metal layer and one vacuum layer. We can converge this in both the muffin-tin potential and in the full potential, and the results are more or less the same when it comes to quantities like density of states and charge-density. For the muffin-tin potential, the potential in the vacuum layers varies only with z, see (8.4). However, the workfunction differs considerably: for the muffin-tin surface the work-function is 5.6 eV, whereas for the full potential surface it is 4.4 eV. We can explain this difference by the fact that the muffin-tin potential is not a very good description for the metal to vacuum interface region. The muffin-tin form of the potential works well when the atoms have a high point-group symmetry, but at the surface this is no longer the case and it gives a rather poor description of the surface potential. Are the physical quantities which we have calculated converged, or have we introduced rather strange results by forcing the potential to be completely bulk-like after only one atomic layer? To check this, we have to include more layers in the surface region.

11.1.2.2 Using 3 sub-volumes

The next surface region we can try is one with two metal subvolumes and one vacuum subvolume. When we converge the potential, using both muffin-tin and full potentials, we get work-functions of 5.7 eV and 4.4 eV respectively. We see that the work-functions are more or less converged with the number of layers. The 4.4 eV value for the full potential calculation

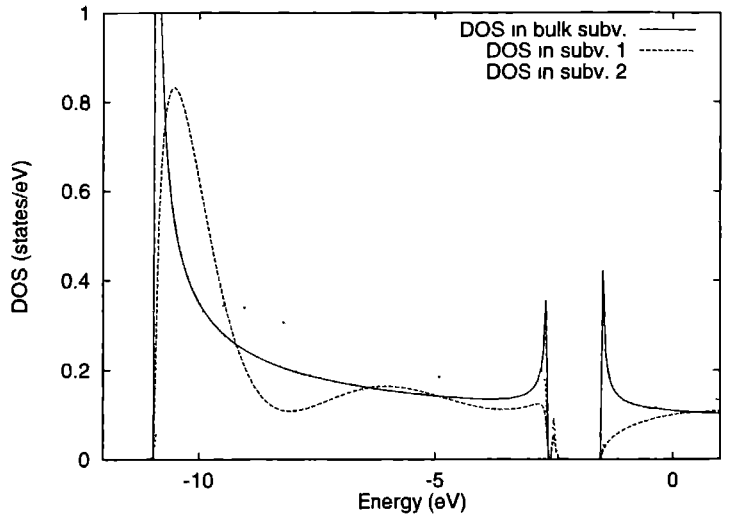


Figure 11.4: *Densities of states for the different subvolumes. Subvolume 2 is the surface layer and subvolume 1 is the adjacent atomic layer. The energy-zero is the Fermi-energy. The imaginary energy is chosen at* 1.0×10^{-5} *Hartree.*

is rather close to the experimental value (of 4.41 eV[32]). This value is probably fortuitously close, since the local density approximation doesn't really work in the surface to vacuum region where the charge-density becomes very small.

We can now compare densities of states in the bulk with the densities of states in the surface subvolumes. We pick the $\mathbf{k}_{\|}$-point $\bar{\Gamma}$, for which the results are shown in figure 11.4. In both the surface and subsurface subvolumes the band edges behave like $\sqrt{E}$[33], though the subsurface behaviour is beginning to resemble the bulk. In the bulk band-edges behave like $\frac{1}{\sqrt{E}}$ which is clearly visible in the figure. The fact that the peaks do not go to infinity is a direct result of the energy-mesh chosen to generate these DOS and the fact that we have a small but finite imaginary energy. If we look very closely, we can see a state within the band-gap. In figure 11.5 I have blown up the region of the band-gap, and we can clearly see a surface state. In our method it is indeed very easy to identify this state as a surface state, because the state clearly exists in a gap in the bulk continuum. Unlike in slab calculations in which all states are discrete, our method clearly distinguishes between bulk and surface states without any complicated - or arbitrary - analysis.

11.1.2.3 Should we use more subvolumes?

In principle we could increase the number of metal subvolumes in the surface region to three and more, but the results shown in the previous section are converged when it comes to the number of layers in which we allow the potential to change from the bulk potential.

We can understand this from a screening argument. In metals the screening length is typically one or two atomic layers, so the effects of the surface are screened away after two

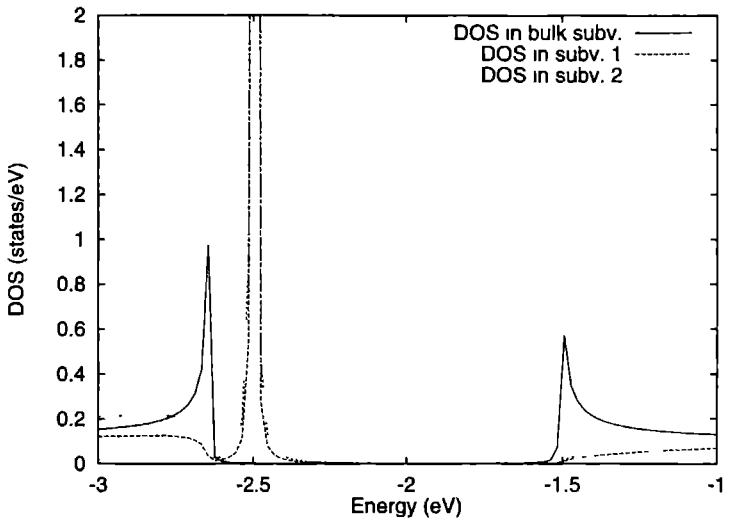


Figure 11.5: *Densities of states for the different subvolumes, for a smaller energy range. The imaginary energy is moved up to* 1.0×10^{-4} *to make the surface state better visible.*

layers. Therefore the potential in the third layer will be bulk-like. To show the effect of screening, I have included the charge-density of the three subvolume calculation (two subvolumes metal, one subvolume vacuum) in figure 11.6. From figure 11.6 it is clear that the embedding framework put into the computer code works, because we cannot see where the boundaries between subvolumes are. In this figure there are actually two such boundaries – the left-hand part of the figure is in subvolume 1 and the right-hand part is in subvolume 3, the vacuum subvolume. The boundary between subvolumes 1 and 2 is at $z = 2.8$ and the boundary between subvolumes 2 and 3 is at $z = 7.4$.

Since the aluminium surface is only a test case, we shall not perform a detailed analysis of the surface and its surface states, but continue with the next test.

11.2 The Cu(100) test surface

The next test is Cu(100), with the same structure as Al(100). We have chosen copper as the next test, because whereas aluminium only has s and p states, copper also has non-trivial d states. So this is a good test as to whether the program can handle d-bands correctly.

11.2.1 Bulk Cu

The structure of Cu is also face-centered-cubic, but with a different lattice constant. We again choose our subvolumes perpendicular to the (100) direction. Converging the bulk, first using the muffin-tin potential and then the full potential, gives us the band-structure and density of states shown in figure 11.7, for the $\mathbf{k}_{\parallel} = (0,0)$. The band-structure has been compared to

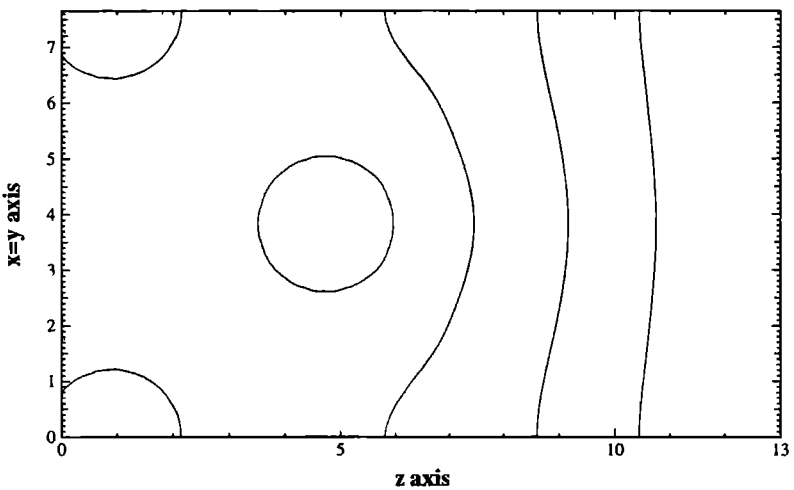


Figure 11.6: *Charge-density contours on a plane defined by a line along $x = y$ in the direction parallel to the subvolume and a line along the z-direction. The contours are exponential. It is clear that the charge on the surface atoms is not spherically symmetric and has a clear distortion in the direction of the next nearest neighbour in the next layer.*

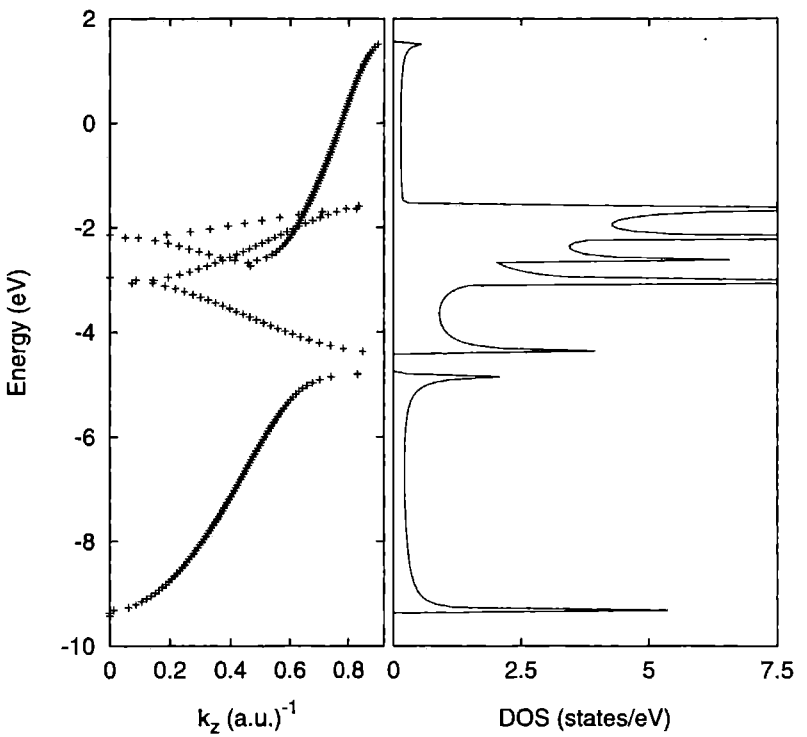


Figure 11.7: *Band structure and density of states of Cu for the* $\bar{\Gamma}$ *point of the 2D Brillouin zone,* $\mathbf{k}_{||} = (0,0)$. *The imaginary energy for the DOS is* 1.0×10^{-5} *Hartree. The zero of energy is the Fermi energy.*

orbital	bulk subv. (%)	surface subv. (%)
s	19.2	0.7
p_x	0.0	0.0
p_y	0.0	0.0
p_z	30.3	0.3
d_{xy}	1.4	0.1
d_{xz}	5.9	48.7
d_{yz}	5.9	48.7
$d_{x^2-y^2}$	0.0	0.0
d_{z^2}	37.3	1.4

Table 11.1: *Percentage of orbitals in DOS integrated around the lower lying Cu surface state feature at $E \approx -1.48eV$. Note that percentages do not add to 100 because of rounding.*

other calculations[34] and is in excellent agreement.

By putting the band-structure and density of states next to each other as in figure 11.7 we can clearly see that the $\frac{1}{\sqrt{E}}$ band-edge singularities show up precisely where the band-structure has zero slope, as we would expect.

The Cu bulk potential can now be used to calculate the boundary conditions on the bulk side of the Cu surface calculation, and the Green function provides us with the embedding potential.

11.2.2 Cu(100) surface

We treat the Cu(100) surface in the same way as the Al surface. First we calculate the surface potential using two subvolumes of which one is a metal subvolume. The workfunctions found for this surface are 6.15 eV (muffin-tin potential) and 4.98 eV (full potential).

Then we use three subvolumes, now with two metal subvolumes. The workfunctions are then 6.16 eV (muffin-tin potential) and 5.03 eV (full potential). The full potential work function is about 0.4 eV higher than the experimental value of 4.59 eV [32]. How much this is due to the use of LDA, systematic errors in theory or experiment, etc. is not at all clear.

As there are many surface features in the density of states we shall concentrate on a fairly narrow energy range which shows two surface states split from the bulk band edge (figure 11.8). Although they overlap with a bulk continuum, the surface states are sharp and do not hybridize with the bulk. This is because of symmetry as we shall now see.

Using the angular-momentum resolved muffin-tin density of states, we can check the character of each feature and compare to the bulk continuum. We do this by simply integrating the DOS around the surface feature, both in the bulk subvolume and in subvolume 2 of the surface calculation. We shall examine the fraction of the different orbitals in the DOS around each feature. The results for the lowest lying feature in figure 11.8 are listed in table 11.1, and for the higher lying feature in table 11.2. From the percentages listed in table 11.1 we see that the bulk continuum for that energy-range is mostly s, p_z and d_{z^2} character, whereas the surface feature is mainly d_{xz} and d_{yz} character. So this state does not hybridize with the bulk continuum and stays a sharp surface state.

The same is true for the second feature. From the numbers in table 11.2 it is clear that

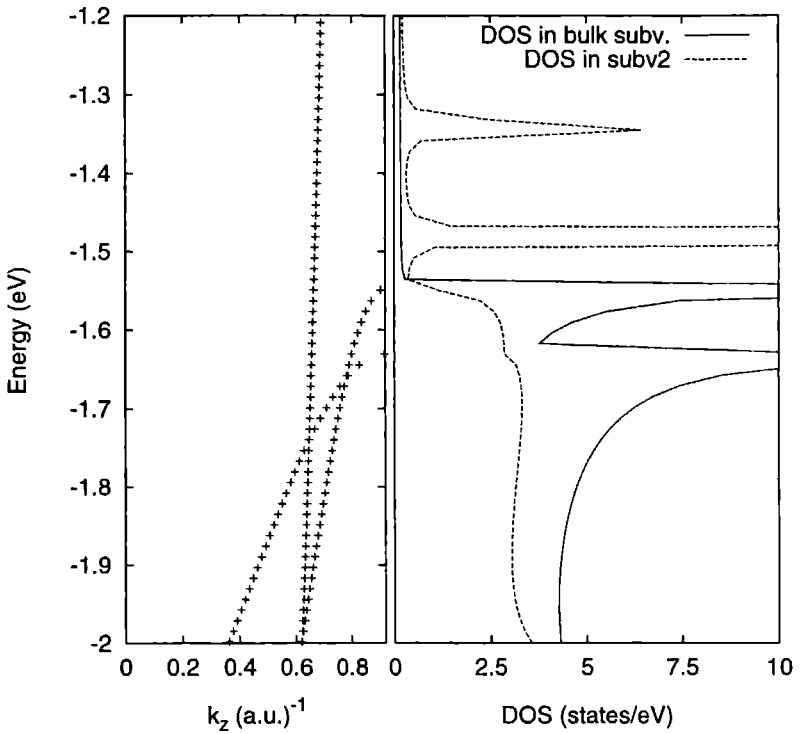


Figure 11.8: *Bulk band-structure and density of states of Cu for the $\bar{\Gamma}$ point of the 2D Brillouin zone, at an energy range near the Fermi energy. An imaginary energy of* 1.0×10^{-5} *Hartree has been used.*

orbital	bulk subv. (%)	surface subv. (%)
s	22.1	5.1
p_x	0.0	0.0
p_y	0.0	0.0
p_z	37.2	3.4
d_{xy}	0.6	81.8
d_{xz}	0.4	0.7
d_{yz}	0.4	0.7
$d_{x^2-y^2}$	0.0	0.0
d_{z^2}	39.0	8.4

Table 11.2: *Percentage of orbitals in DOS integrated around the higher lying surface feature at* $E \approx -1.35$*eV. Note that percentages do not add to 100 because of rounding.*

this feature has mainly d_{xy} character (i.e. in plane) and therefore cannot hybridize with the mainly z character of the orbitals of the bulk states. So the second feature is also a surface state and not a resonance.

Chapter 12

Conductance of magnetic domain walls

In this chapter, I shall propose a new use for the method. We use the transfer matrix and spinor theory to construct a formulation which enables us to examine the conductance of magnetic systems where the magnetisation varies from layer to layer in a certain controlled fashion. We shall apply this to examine the conductance through magnetic domain walls, sometimes called *Bloch* walls[35]. But first we have to look at magnetic ordering and define what a domain wall is.

12.1 Magnetic ordering

We can think of a magnetic material as a distribution of local magnetic moments. This is schematically shown in figure 12.1 for a disordered system. When the local magnetisations or spins are ordered, either through outside means, e.g. an external magnetic field, or through internal means, e.g. exchange interaction, we talk about magnetic ordering. The simplest of these orderings is ferromagnetic ordering, in which all spins prefer to line up in one direction, as shown in figure 12.2. This ordering can be modeled by the ferromagnetic Heisenberg

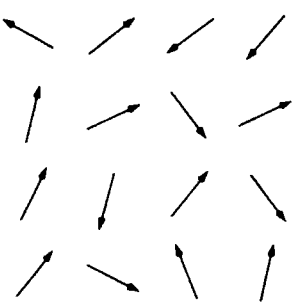


Figure 12.1: *Schematic picture of spins in a disordered magnetic material.*

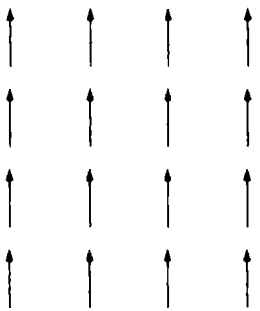


Figure 12.2: *Schematic picture of spins in a ferromagnetically ordered magnetic material.*

model[2]:

$$\mathcal{H} = -\tfrac{1}{2} \sum_{\mathbf{R}\mathbf{R}'} \mathbf{S}(\mathbf{R}) \cdot \mathbf{S}(\mathbf{R}') J(\mathbf{R} - \mathbf{R}') - g\mu_B H \sum_{\mathbf{R}} S_z(\mathbf{R}) \tag{12.1}$$

with

$$J(\mathbf{R} - \mathbf{R}') = J(\mathbf{R}' - \mathbf{R}) \geq 0 \tag{12.2}$$

This Hamiltonian is valid for a ferromagnetic system, because the positive exchange coupling J of (12.2) favours parallel spin alignment.

The Hamiltonian of (12.1) does not include effects due to magnetic dipolar coupling between the moments. The direct dipolar interaction energy of two magnetic dipoles $\mathbf{m}_1$ and $\mathbf{m}_2$ separated by $\mathbf{r}$ is:

$$U = \frac{1}{r^3} \left[\mathbf{m}_1 \cdot \mathbf{m}_2 - 3(\mathbf{m}_1 \cdot \hat{\mathbf{r}})(\mathbf{m}_2 \cdot \hat{\mathbf{r}}) \right] \tag{12.3}$$

In [2] it is derived that the size of U is typically of the order of 10^{-4} eV, which is several orders of magnitude smaller than the energy differences due to the exchange coupling.

The ground state of (12.1) can be shown to be a state in which all magnetic dipoles line up in the direction of the local field $\mathbf{H}$ which defines the z-axis.

It can also be shown[2] that a ferromagnet will exhibit a spontaneous magnetisation even when no field is applied. The magnetic susceptibility governing this magnetisation follows the Curie-Weiss Law and gives us a critical temperature below which the material is magnetised. This critical temperature is known as the *Curie temperature.*

12.1.1 Consequence of dipolar interactions in ferromagnets

So, according to the ferromagnetic Heisenberg Hamiltonian and the Curie-Weiss Law, a piece of iron below its Curie temperature should be magnetised. However, the Curie temperature of iron is over 1000K, and a piece of iron is in general not magnetised. Yet, it can be magnetised rather easily with a "permanent magnet" and is attracted by a magnetic field far too strongly to be a paramagnet.

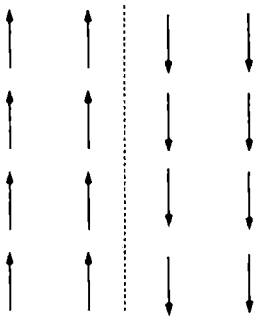


Figure 12.3: *Schematic picture of spins in a ferromagnetically ordered magnetic material, showing two domains.*

The explanation for this apparent inconsistency is to reconsider the magnetic dipolar interaction between the spins. Although as mentioned in the previous section this interaction is rather weak compared to the exchange coupling (typically a thousand times smaller), the dipolar interaction is long-ranged, whereas the exchange coupling is short-ranged. This means that the magnetic order on a larger scale can be quite complex, since the dipolar interaction can become important when many spins are involved.

When we re-examine the uniformly magnetised configuration we have used to describe the ferromagnet (figure 12.2), we see that the dipolar interaction energy for that system can become rather large and this configuration is therefor uneconomical on a macroscopic scale. The dipolar energy can be reduced by dividing the sample up into uniformly magnetised *domains* of macroscopic size, whose magnetisation vectors point in widely different directions (see figure 12.3). This subdivision will cost exchange energy, since the spins near the boundary of a domain will have an unfavourable exchange coupling with the spins in the neighbouring opposite aligned domain. However, the exchange coupling is short-ranged, so only the spins near the boundary between two domains will have their exchange energies raised. This means that we could call the extra exchange energy due to the boundary between domains an interface property. In contrast the dipolar interaction is a bulk property, so the formation of domains means that the dipolar interaction energy of *each* spin will decrease. So if the ratio between domain volume and domain boundary area is large enough, domains will form in spite of the extra exchange energy due to the domain boundaries.

The formation and movement of the boundaries between domains, called *domain walls* or *Bloch walls* is responsible for the way in which a ferromagnet can lose or retain its spontaneous magnetisation below the Curie temperature.

In the next section we shall look at domain walls in more detail.

12.2 What is a domain wall?

As we have seen, a macroscopic ferromagnet favours the formation of domains, separated by domain walls. But what is the exact nature of these walls? A first approximation is the abrupt wall, sketched in the top part of figure 12.4. This type of wall is unnecessarily costly in exchange energy. By spreading the reversal of the spin over many spins, as sketched

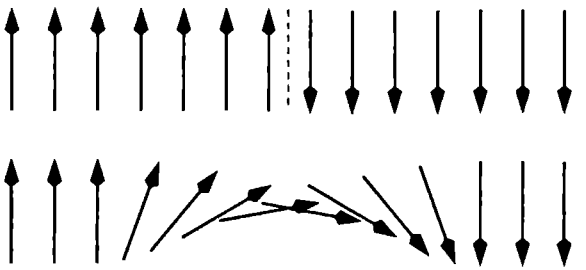


Figure 12 4 *Sketch of the behaviour of spins in a domain wall, showing (top) an abrupt boundary and (bottom) a gradual boundary The latter is less costly in exchange energy*

in the bottom part of figure 12 4, the exchange energy of the domain wall can be lowered substantially If the spin reverses over n spins (n substantially large[1], i e ≥ 10), then each spin will be mismatched in orientation with its neighbours over an angle of $\frac{\pi}{n}$ This means that in a classical approximation the exchange energy of the two neigbouring spins is not the minimum value $-JS^2$, but $-JS^2\cos(\frac{\pi}{n}) \approx -JS^2\left[1-\frac{1}{2}\left(\frac{\pi}{n}\right)^2\right]$ Thus the energy of spin reversal over n spins is[2]

$$\Delta E = n\left[-JS^2\cos\left(\frac{\pi}{n}\right)-(-JS^2)\right] = \frac{\pi^2}{2n}JS^2 \qquad (12\ 4)$$

So for n large enough (which is necessary for this simple model to hold), this is smaller than the exchange energy of the abrupt domain wall by a factor $\frac{\pi^2}{2n}$ So if the exchange energy of the domain wall is the only consideration, the domain wall would grow to a thickness limited only by dipolar interactions We have assumed in the simple model above that the exchange interaction between neighbouring spin is isotropic, depending only on the angle between them This is true, however, only because we neglected spin-orbit interactions In a real crystal the spins are coupled to the lattice through the spin-orbit interaction This means that the energy depends also on the absolute orientation of the spins with respect to the crystal axes as well as the relative orientation with respect to other spins This dependence on absolute orientation, known as the *anisotropy energy*[2, 36], is quite weak, but is strong enough to add a fixed energy to each spin in the line of spins in the domain wall Therefore it will eventually prevent the domain wall from growing thicker[37, 36] Thus the thickness of the domain wall is determined by a balance between the exchange and anisotropy energies

In this chapter we shall concentrate on domain walls between very large domains with a given magnetisation These domain walls are of a given thickness, and we shall only examine domain walls whose boundaries are flat, i e form a plane This enables us to use a layered description for the domain wall The two semi-infinite domains are treated using the bulk mode of the program This gives us the transfer matrix for these domains In between the two domains we have a variable number of layers, N, which have a direction of magnetisation varying linearly between the directions in the two semi-infinite domains A system which

[1]The thickness of the domain wall should not be so large however, that the dipolar energy of the boundary itself is significant

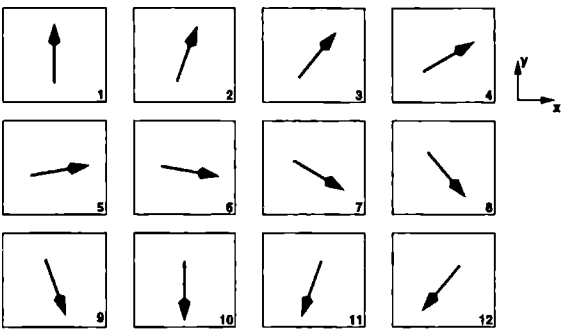


Figure 12.5: *View of the unit cell in 12 subvolumes (numbered in lower right corners). In each subvolume the arrow indicates the direction of the magnetisation in that subvolume. The in-plane coordinates are indicated to the right of the views of the subvolumes.*

has the direction of the magnetisation changing from one atomic layer to the next, always rotating over a fixed angle $\Delta\theta$, we will call a spin-spiral.

12.3 The spin-spiral system

Before we look at the domain-walls, we shall first examine the bulk spin-spiral systems. When the subvolumes in figure 12.5 are stacked, we obtain a spin spiral. We define a spin-spiral system as a stack of subvolumes in which the magnetisation in successive subvolumes is rotated through a constant angle $\Delta\theta$.

These spin-spiral systems form a whole new subset of materials. The first step we take towards performing calculations on domain walls is to calculate the electronic structure of a bulk spin-spiral system consisting of an *infinite* stack of subvolumes.

12.3.1 Theory of spin-spiral systems

In our approach to the spin-spiral, we use the same techniques which have been used before to handle systems with helical or spiral magnetisation behaviour [38, 39, 40].

The first step is to define the spinor rotation matrix $U(\theta, \mathbf{u})$. This matrix rotates the spin degrees of freedom over an angle θ around the axis $\mathbf{u}$. We use the following definition[13]:

$$U(\theta, \mathbf{u}) = \begin{pmatrix} \cos\frac{\theta}{2} - iu_z \sin\frac{\theta}{2} & (-iu_x - u_y)\sin\frac{\theta}{2} \\ (-iu_x + u_y)\sin\frac{\theta}{2} & \cos\frac{\theta}{2} + iu_z \sin\frac{\theta}{2} \end{pmatrix} \tag{12.5}$$

where u_x, u_y and u_z are the cartesian components of the vector $\mathbf{u}$.

In [39] it is shown that the Hamiltonian of a system in which the magnetisation is rotated by an angle θ around $\mathbf{u}$ is given by:

$$H_{\mathbf{m}} = -\tfrac{1}{2}\nabla^2 + U^{\dagger}(\theta, \mathbf{u}) \begin{pmatrix} V_{\mathrm{up}}(\mathbf{r}) \\ V_{\mathrm{down}}(\mathbf{r}) \end{pmatrix} U(\theta, \mathbf{u}) \tag{12.6}$$

where the operator $-\frac{1}{2}\nabla^2$ can be taken outside of the spinor rotation operators because the operator is isotropic in spin-space. Now the Hamiltonian with magnetisation along the z-axis is given (10.9) by:

$$H_0 = \begin{pmatrix} H_{uu} & 0 \\ 0 & H_{dd} \end{pmatrix}, \tag{12.7}$$

hence (12.6) becomes:

$$H_{\mathbf{m}} = U^{\dagger}(\theta, \mathbf{u}) H_0 U(\theta, \mathbf{u}) = U^{\dagger}(\theta, \mathbf{u}) \begin{pmatrix} H_{uu} & 0 \\ 0 & H_{dd} \end{pmatrix} U(\theta, \mathbf{u}) \tag{12.8}$$

Since the overlap matrix is diagonal in spin-space, we also get:

$$H_{\mathbf{m}} - \mathcal{E}O = U^{\dagger}(\theta, \mathbf{u}) \left(H_0 - \mathcal{E}O\right) U(\theta, \mathbf{u}) \tag{12.9}$$

This means for the Green function:

$$\begin{aligned} G_{\mathbf{m}} &= (H_{\mathbf{m}} - \mathcal{E}O)^{-1} \\ &= \left[U^{\dagger}(\theta, \mathbf{u}) \left(H_0 - \mathcal{E}O\right) U(\theta, \mathbf{u})\right]^{-1} \\ &= U(\theta, \mathbf{u})^{-1} \left(H_0 - \mathcal{E}O\right)^{-1} U^{\dagger}(\theta, \mathbf{u})^{-1} \\ &= U^{\dagger}(\theta, \mathbf{u}) G_0 U(\theta, \mathbf{u}) \end{aligned} \tag{12.10}$$

From the way that the scattering properties depend on the Green function (see section 3.4.1) and the form of the transfer matrix (see section 3.4.2.2), we can derive the following expression:

$$\mathbb{T}^{\rightarrow}_{\mathbf{m}} = U^{\dagger}(\theta, \mathbf{u}) \begin{pmatrix} \mathbb{T}^{\rightarrow}_{0,uu} & 0 \\ 0 & \mathbb{T}^{\rightarrow}_{0,dd} \end{pmatrix} U(\theta, \mathbf{u}) \tag{12.11}$$

where the blocks in the transfer matrix in the right-hand part of (12.11) are the transfer matrices for spin-up and spin-down in the normal bulk of the material forming the spin-spiral. Now that we have derived the necessary mathematics, let us examine the exact situation we want to calculate.

To find the conductance through a piece of bulk spin-spiral, we have to construct the Bloch states of the system. To do this we have to use a modified Bloch's theorem:

$$\mathbb{T}^{\rightarrow}_{\mathbf{m}} U(\Delta\theta, \mathbf{u}) b_{\mathbf{k}}(\mathbf{r}) = e^{i\mathbf{k}\,\mathbf{r}} b_{\mathbf{k}}(\mathbf{r}) \tag{12.12}$$

where the extra $U(\Delta\theta, \mathbf{u})$ arises because of the spiral nature of the system. The vector $\mathbf{k}$ no longer is a wave vector for a pure translation, but contains a certain rotation part.

Let us examine what this means for the band-structure, using a free-electron system as an example. We define a vector $\mathbf{q}$ such that $\mathbf{q} \cdot \mathbf{r}_t = \Delta\theta$, where $\mathbf{r}_t$ is the vector taking us from the coordinate centre of one subvolume in the spin-spiral to the coordinate centre in the next subvolume. This means that we can write the spinor rotation on the left-hand side of (12.12) as:

$$U(\Delta\theta, \mathbf{u}) = \begin{pmatrix} e^{-\frac{1}{2}i\mathbf{q}\,\mathbf{r}_t} & 0 \\ 0 & e^{\frac{1}{2}i\mathbf{q}\,\mathbf{r}_t} \end{pmatrix} \tag{12.13}$$

Substituting this into (12.12) for a free electron $\mathbb{T}^{\rightarrow}_{\mathbf{m}}$, we find for the Bloch states:

$$b_{\mathbf{k}}(\mathbf{r}) = \begin{pmatrix} e^{i(\mathbf{k}-\frac{1}{2}\mathbf{q})\cdot\mathbf{r}} \\ e^{i(\mathbf{k}+\frac{1}{2}\mathbf{q})\cdot\mathbf{r}} \end{pmatrix} \tag{12.14}$$

This means that the band-structure splits up into spin-up bands with wave-vector $\mathbf{k}-\frac{1}{2}\mathbf{q}$ and spin-down bands with wave-vector $\mathbf{k}+\frac{1}{2}\mathbf{q}$.

For a non-free-electron system, we get the same wave-vector splitting, but now the spin-up and spin-down bands will hybridize because there is an interaction between spin-up and spin-down. An example of this is shown in figure 12.6, which shows the band-structure of a Ni(100) spin-spiral compared to the bulk band-structure of the same material. We see that the wave-vectors are split by a distance q_z. Also the band-structure on the right is not just the band-structure on the left with this wave-vector split. We notice hybridization around $k_z = 0$, both in the *sp*-bands at lower energies and in the *d*-bands just below the Fermi energy.

Once we have found the Bloch states for the spin-spiral system, and the basis transformation matrix $\mathbb{U}$ (see section 4.2.2) corresponding to them, we can use these to try and tackle the domain wall system.

12.4 Using the spin-spiral for the domain wall

Schematically the system under investigation is shown in figure 12.7, where the direction of magnetisation is depicted as rotating out of the plane of the layers for clarity, though in practice we rotate the magnetisation in the plane of the layers. In our model we take the angle $\Delta\theta$ between layers to be constant. This means that the direction of the magnetisation varies linearly with the layer. This is an approximation we need to make to be able to solve the problem.

Now the problem that we need to solve is to find a transfer matrix for the stack of subvolumes starting in the left ferromagnet Fm1 and ending in the right ferromagnet Fm2. We solved a similar problem in chapter 4, where we wanted to find the conductance through an interface (see section 4.4).

The approach to this problem is similar. We assume we have a state approaching the system from the left. This consists of a Bloch state of the Fm1 ferromagnet. The first step is to let the basis transformation matrix $\mathbb{U}_{\mathrm{Fm1}}$ act on the state, to get the expansion of the state in terms of the internal basis functions of the embedding/conductance code. Then we act on the state with the transfer matrix of the Fm1 ferromagnet, $\mathbb{T}^{\rightarrow}_{\mathrm{Fm1}}$, to end up with the state on the boundary between Fm1 and the spin-spiral. To transfer the state through the spin-spiral, we have to let the transfer matrices of all the subvolumes in the spiral act on our state. The transfer matrix in subvolume n can be written as:

$$\mathbb{T}^{\rightarrow}_{\mathrm{Sp},n} = U^{\dagger}(n(\Delta\theta),\mathbf{u})\mathbb{T}^{\rightarrow}_{\mathrm{Fm}}U(n(\Delta\theta),\mathbf{u}) \tag{12.15}$$

where $\mathbb{T}^{\rightarrow}_{\mathrm{Fm}}$ is the transfer matrix for one subvolume of the material of the spin-spiral without any spiral behaviour. Once we end up with the state on the boundary between the spin-spiral and Fm2, we apply the transfer matrix for one subvolume of Fm2, $\mathbb{T}_{\mathrm{Fm2}}$, to end up with the state in the right ferromagnet in term of the internal basis. All that is left is to basis transform to the Bloch basis in the right ferromagnet to find the state in terms of the travelling Bloch

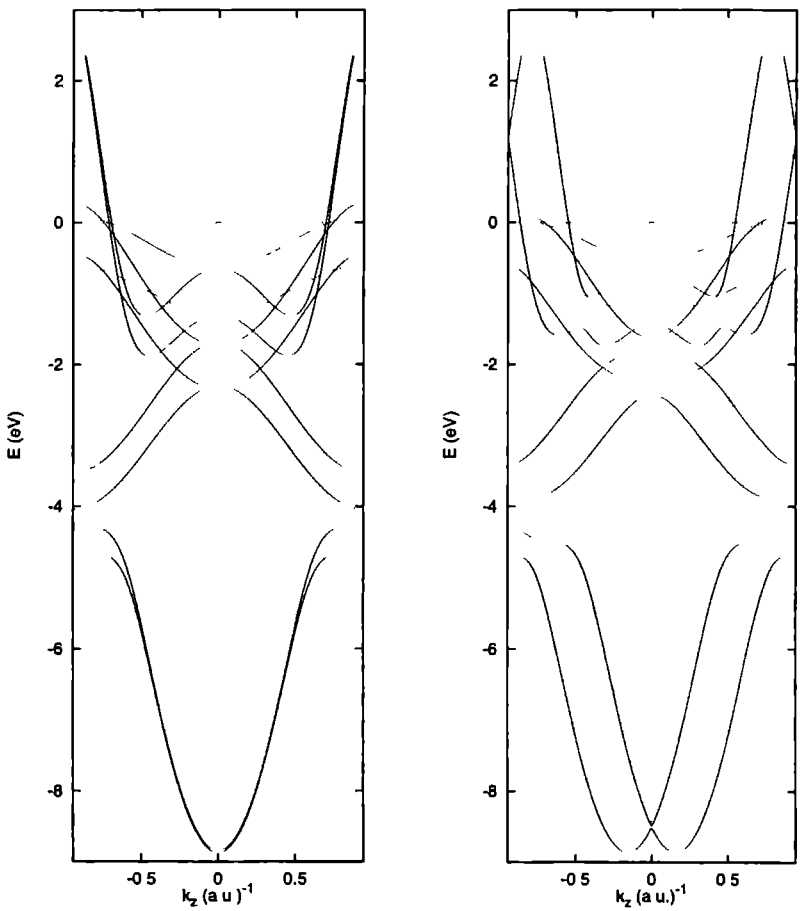


Figure 12.6: *Band-structure of a stacking of Ni(100) layers. (left): bulk material. (right): spin-spiral with an angle of* 60° *between the layers.*

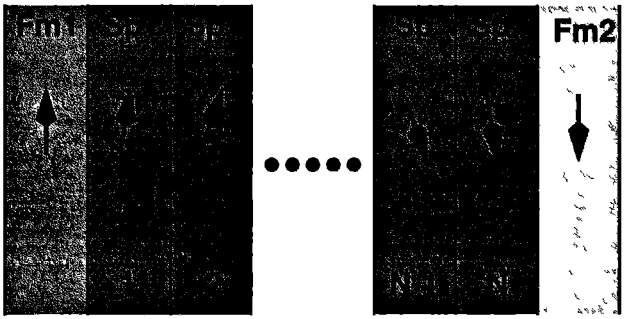


Figure 12.7: *Schematic picture of the domain wall.* ***Fm1*** *indicates the ferromagnet in the left semi-infinite domain, and* ***Fm2*** *indicates the ferromagnet in the right semi-infinite domain.* ***Sp*** *indicates the spin-spiral layers, each with a magnetisation rotated over $\Delta\theta$ with respect to the previous layer.*

states in Fm2. So the total transfer matrix for the stack in figure 12.7 in terms of Bloch states is given by:

$$\mathbb{T}^{\rightarrow}_{\text{Bloch}} = \mathbb{U}^{-1}_{\text{Fm2}}\mathbb{T}^{\rightarrow}_{\text{Fm2}}\mathbb{T}^{\rightarrow}_{\text{Sp},N}\mathbb{T}^{\rightarrow}_{\text{Sp},N-1}\cdots\mathbb{T}^{\rightarrow}_{\text{Sp},2}\mathbb{T}^{\rightarrow}_{\text{Sp},1}\mathbb{T}^{\rightarrow}_{\text{Fm1}}\mathbb{U}_{\text{Fm1}} \tag{12.16}$$

The total number of layers, N, in the domain wall can become quite large, therefore we cannot perform the calculation in (12.16), because a product of too many transfer matrices becomes numerically unstable. But apart from implementation problems, (12.16) in principle gives the exact transfer matrix for the spin-spiral domain wall.

The first step in solving the implementation problem is to rewrite (12.16) using the expression in (12.15):

$$\begin{aligned}\mathbb{T}^{\rightarrow}_{\text{Bloch}} = \mathbb{U}^{-1}_{\text{Fm2}}\mathbb{T}^{\rightarrow}_{\text{Fm2}}U^{\dagger}(N(\Delta\theta),\mathbf{u})\mathbb{T}^{\rightarrow}_{\text{Fm}}U(N(\Delta\theta),\mathbf{u})U^{\dagger}((N-1)(\Delta\theta),\mathbf{u})\times\\ \mathbb{T}^{\rightarrow}_{\text{Fm}}U((N-1)(\Delta\theta),\mathbf{u})\cdots U^{\dagger}(2(\Delta\theta),\mathbf{u})\mathbb{T}^{\rightarrow}_{\text{Fm}}\times\\ U(2(\Delta\theta),\mathbf{u})U^{\dagger}((\Delta\theta),\mathbf{u})\mathbb{T}^{\rightarrow}_{\text{Fm}}U(\Delta\theta,\mathbf{u})\mathbb{T}^{\rightarrow}_{\text{Fm1}}\mathbb{U}_{\text{Fm1}}\end{aligned} \tag{12.17}$$

We now use the fact that:

$$U(n(\Delta\theta),\mathbf{u})U^{\dagger}((n-1)(\Delta\theta),\mathbf{u}) = U(\Delta\theta,\mathbf{u}) \tag{12.18}$$

to write (12.17) as:

$$\begin{aligned}\mathbb{T}^{\rightarrow}_{\text{Bloch}} = \mathbb{U}^{-1}_{\text{Fm2}}\mathbb{T}^{\rightarrow}_{\text{Fm2}}U^{\dagger}(N(\Delta\theta),\mathbf{u})\mathbb{T}^{\rightarrow}_{\text{Fm}}U(\Delta\theta,\mathbf{u})\mathbb{T}^{\rightarrow}_{\text{Fm}}U(\Delta\theta,\mathbf{u})\cdots\\ \mathbb{T}^{\rightarrow}_{\text{Fm}}U(\Delta\theta,\mathbf{u})\mathbb{T}^{\rightarrow}_{\text{Fm}}U(\Delta\theta,\mathbf{u})\mathbb{T}^{\rightarrow}_{\text{Fm1}}\mathbb{U}_{\text{Fm1}}\end{aligned} \tag{12.19}$$

We now define:

$$\widetilde{\mathbb{T}^{\rightarrow}_{\text{Fm}}} = \mathbb{T}^{\rightarrow}_{\text{Fm}}U(\Delta\theta,\mathbf{u}) \tag{12.20}$$

to obtain:

$$\mathbb{T}^{\rightarrow}_{\text{Bloch}} = \mathbb{U}^{-1}_{\text{Fm2}}\mathbb{T}^{\rightarrow}_{\text{Fm2}}U^{\dagger}(N(\Delta\theta),\mathbf{u})\left(\widetilde{\mathbb{T}^{\rightarrow}_{\text{Fm}}}\right)^{N}\mathbb{T}^{\rightarrow}_{\text{Fm1}}\mathbb{U}_{\text{Fm1}} \tag{12.21}$$

The quantity defined in (12.20) is the transfer matrix used in (12.12) to find the Bloch states for the spin-spiral. This means that we can write:

$$\mathbb{T}^{\rightarrow}_{\text{Bloch}} = \mathbb{U}^{-1}_{\text{Fm2}}\mathbb{T}^{\rightarrow}_{\text{Fm2}}U^{\dagger}(N(\Delta\theta),\mathbf{u})\widetilde{\mathbb{T}^{\rightarrow}_{\text{Fm}}}\mathbb{U}_{\text{Fm}}\mathbb{U}^{-1}_{\text{Fm}}\widetilde{\mathbb{T}^{\rightarrow}_{\text{Fm}}}\mathbb{U}_{\text{Fm}}\mathbb{U}^{-1}_{\text{Fm}}\cdots \\ \mathbb{U}_{\text{Fm}}\mathbb{U}^{-1}_{\text{Fm}}\widetilde{\mathbb{T}^{\rightarrow}_{\text{Fm}}}\mathbb{U}_{\text{Fm}}\mathbb{U}^{-1}_{\text{Fm}}\widetilde{\mathbb{T}^{\rightarrow}_{\text{Fm}}}\mathbb{T}^{\rightarrow}_{\text{Fm1}}\mathbb{U}_{\text{Fm1}} \quad (12.22)$$

We can now rewrite this in the form:

$$\mathbb{T}^{\rightarrow}_{\text{Bloch}} = \mathbb{U}^{-1}_{\text{Fm2}}\mathbb{T}^{\rightarrow}_{\text{Fm2}}U^{\dagger}(N(\Delta\theta),\mathbf{u})\widetilde{\mathbb{T}^{\rightarrow}_{\text{Fm}}}\mathbb{U}_{\text{Fm}}\left[\mathbb{U}^{-1}_{\text{Fm}}\widetilde{\mathbb{T}^{\rightarrow}_{\text{Fm}}}\mathbb{U}_{\text{Fm}}\right]^{N-2}\mathbb{U}^{-1}_{\text{Fm}}\widetilde{\mathbb{T}^{\rightarrow}_{\text{Fm}}}\mathbb{T}^{\rightarrow}_{\text{Fm1}}\mathbb{U}_{\text{Fm1}} \quad (12.23)$$

This expression can be split into three parts. First we have the transfer matrix from Bloch states in Fm1 to Bloch states in the spin-spiral for the Fm1 to spin-spiral interface:

$$\mathbb{T}^{\rightarrow}_{\text{Bloch,Fm1/Sp}} = \mathbb{U}^{-1}_{\text{Fm}}\widetilde{\mathbb{T}^{\rightarrow}_{\text{Fm}}}\mathbb{T}^{\rightarrow}_{\text{Fm1}}\mathbb{U}_{\text{Fm1}} \quad (12.24)$$

Secondly we have the transfer matrix through $N-2$ layers of spin-spiral:

$$\mathbb{T}^{\rightarrow}_{\text{Bloch,Sp,N-2}} = \left[\mathbb{U}^{-1}_{\text{Fm}}\widetilde{\mathbb{T}^{\rightarrow}_{\text{Fm}}}\mathbb{U}_{\text{Fm}}\right]^{N-2} \quad (12.25)$$

This transfer matrix is diagonal, which leads to our method being computationally independent of the domain wall thickness N. And thirdly we have the transfer matrix for the interface between the spin-spiral and Fm2 in terms of Bloch states of the spin-spiral into Bloch states of Fm2:

$$\mathbb{T}^{\rightarrow}_{\text{Bloch,Sp/Fm2}} = \mathbb{U}^{-1}_{\text{Fm2}}\mathbb{T}^{\rightarrow}_{\text{Fm2}}U^{\dagger}(N(\Delta\theta),\mathbf{u})\widetilde{\mathbb{T}^{\rightarrow}_{\text{Fm}}}\mathbb{U}_{\text{Fm}} \quad (12.26)$$

Then (12.23) becomes:

$$\mathbb{T}^{\rightarrow}_{\text{Bloch}} = \mathbb{T}^{\rightarrow}_{\text{Bloch,Sp/Fm2}}\mathbb{T}^{\rightarrow}_{\text{Bloch,Sp,N-2}}\mathbb{T}^{\rightarrow}_{\text{Bloch,Fm1/Sp}} \quad (12.27)$$

These transfer matrices can now be limited to just the travelling Bloch states. This leads to problems, however, when the number of travelling Bloch states in the spin-spiral is smaller than in the ferromagnets. The resulting transfer matrix is then singular, which means that it is impossible to find the transmission matrix and to calculate the conductance. To tackle this problem we use multiple-scattering methods.

12.4.1 Multiple scattering in the domain wall

From the transfer matrices defined in (12.24), (12.25) and (12.26) we can construct the scattering matrices for just the travelling Bloch states. A scattering matrix is defined as:

$$S = \begin{pmatrix} R_L & T_R \\ T_L & R_R \end{pmatrix} \quad (12.28)$$

Let us now introduce the following notation:

$$S_{\text{Fm1/Sp}} = \begin{pmatrix} R^1_L & T^1_R \\ T^1_L & R^1_R \end{pmatrix} \quad (12.29)$$

$$S_{\text{Sp}} = \begin{pmatrix} R^S_L & T^S_R \\ T^S_L & R^S_R \end{pmatrix} \quad (12.30)$$

$$S_{\text{Sp/Fm2}} = \begin{pmatrix} R^2_L & T^2_R \\ T^2_L & R^2_R \end{pmatrix} \quad (12.31)$$

We can now construct the transmission and reflection matrices for the entire domain wall, by using the transmission and reflection properties of the separate building blocks. By using multiple scattering we obtain the following expressions for the scattering properties of the domain wall (DW).

$$T_L^{\mathrm{DW}} = T_L^2 T_L^S \left[I - R_R^1 T_R^S R_L^2 T_L^S\right]^{-1} T_L^1 \tag{12.32a}$$

$$T_R^{\mathrm{DW}} = T_R^1 T_R^S \left[I - R_L^2 T_L^S R_R^1 T_R^S\right]^{-1} T_R^2 \tag{12.32b}$$

$$R_L^{\mathrm{DW}} = R_L^1 + T_R^1 \left[I - T_R^S R_L^2 T_L^S R_R^1\right]^{-1} T_R^S R_L^2 T_L^S T_L^1 \tag{12.32c}$$

$$R_R^{\mathrm{DW}} = R_R^2 + T_L^2 \left[I - T_L^S R_R^1 T_R^S R_L^2\right]^{-1} T_L^S R_R^1 T_R^S T_R^2 \tag{12.32d}$$

This gives us the scattering matrix for the domain wall:

$$S^{\mathrm{DW}} = \begin{pmatrix} R_L^{\mathrm{DW}} & T_R^{\mathrm{DW}} \\ T_L^{\mathrm{DW}} & R_R^{\mathrm{DW}} \end{pmatrix} \tag{12.33}$$

which we then use to construct the Bloch transfer matrix for the domain wall, using (3.62). Using the transmission matrix, T_L^{DW}, we can calculate the conductance using the same algorithm as described in section 4.4.

12.4.2 Some implementation details

Since the calculation of the transmission matrix of the domain wall depends only on the transfer matrices of the subvolumes in the different ferromagnets and the spin-spiral (and those are normally the same material), we do not have to perform a full-blown calculation with the `GROW.MAG` module. An adapted `GROW.MAG` module is run, which outputs the spin-up transfer matrices for all $\mathbf{k}_{\|}$ in the file "gtransmat.up" and the spin-down transfer matrices in the file "gtransmat.down".

A special program has been written to perform the domain wall calculation as outlined in the previous section. This program reads in the necessary transfer matrices, which means that it can be quite fast. The output of this `SPIN` program are the conductances for the Fm1 ferromagnet to spin-spiral interface, the spin-spiral bulk, both ferromagnet bulk, the spin-spiral to Fm2 ferromagnet interface, and of course the conductance of the entire domain wall.

Since the method incorporates the thickness, N, of the domain wall in an exponent (see (12.25)), the calculation time does not depend on this thickness. Also the transfer matrices are stored in files, and they are not recalculated again and again as they should have been if the functionality of `SPIN` had been built into `GROW.MAG`.

12.5 Results for different materials

In the general case of a domain wall, the 'bare' transfer matrices for the different shaded areas in figure 12.7 are the same. This means that for every material we have to construct the transfer matrices once.

In this section I briefly show some results for domain walls in the three ferromagnetic materials: Fe, Co, and Ni. The bulk calculations for these materials gave the following magnetic moments: Co (1.72 μB), Fe (2.28 μB), and Ni (0.60 μB).

Some details about the materials and the domain walls in these materials are given in table 12.1[36]. The first thing to notice is that for Fe and especially Ni the domain walls are quite thick, i.e. far removed from the abrupt domain wall discussed earlier. The domain walls for the three materials have been calculated for several thicknesses. We shall look at each material in a little detail in the following sections.

12.5.1 The nickel domain wall

The layers in the Ni domain wall are fcc(100) layers. The transfer matrices have been calculated using the adapted `GROW.MAG` module in bulk mode. The potentials used are muffin-tin potentials converged for both $\mathbf{k}_{\|}$-points and energies in the energy integral. In figure 12.8 I give conductance values as a function of the $\Delta\theta$ angle between the subvolumes in the domain wall. For fixed magnetisation directions θ_1 and θ_2 in the left and right ferromagnets, $\Delta\theta$ is directly linked to the thickness N of the domain wall through:

$$\Delta\theta = \frac{\theta_2 - \theta_1}{N+1} \tag{12.34}$$

In general the angles θ_1 and θ_2 can take any value (see later sections), but for now we take $\theta_1 = 0$ and $\theta_2 = \pi$, which gives us a 180 degree domain wall.

The first thing to notice from figure 12.8 is that in each case the conductance approaches the bulk value for small angles. This is logical, since an angle of zero (thickness infinite) is just the normal bulk situation. Another interesting property is that the conductance of the bulk spin-spiral is smaller than that of the bulk (as expected), but the conductance through the entire domain wall is a lot smaller than the bulk spin-spiral conductance. This can be explained by the fact that the Bloch states coming from the left ferromagnet will reflect off the ferromagnet to spin-spiral interface. So most of the conductance loss occurs at those interfaces.

Since the domain wall for Ni is about 570 subvolumes thick according to table 12.1, which corresponds to a $\Delta\theta \approx 0.32°$, a magnified plot of the conductances for small angles is shown in figure 12.9. From this plot it is less clear that the conductances converge to the same value for small angles, but numerical errors may start to crop up here. The $\mathbf{k}_{\|}$-set used is only 465 vectors in the irreducible part of the two-dimensional Brillouin zone. If we choose the thickness of the domain wall as given in table 12.1, we get a conductance through the domain wall that is 99.85% of the bulk value. This is a decrease in conductance due to the domain wall of just 0.15%, a very small decrease. This would seem to indicate that we are dealing with spin-precession here, in which the spin adiabatically follows the magnetisation as an electron moves across the domain wall. In the more abrupt domain wall, the reflection is

Property	Fe	Co	Ni
Domain wall thickness			
(in nm)	40	15	100
(in lattice parameters)	138	36	285
(in subvolumes)	276	72	570
Curie temperature (in K)	1043	1404	631

Table 12.1: *Selected properties of the materials under investigation[36].*

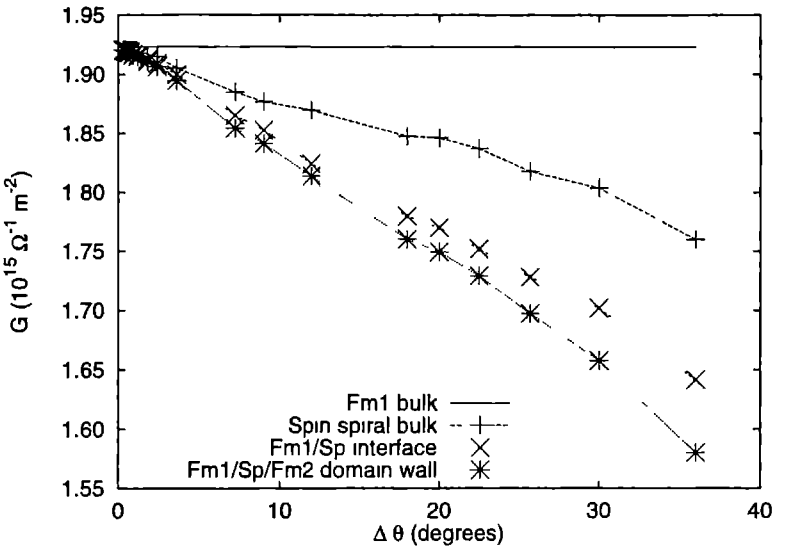


Figure 12.8: *Conductances of the Ni domain wall as a function of the angle, $\Delta\theta$, between subvolumes.*

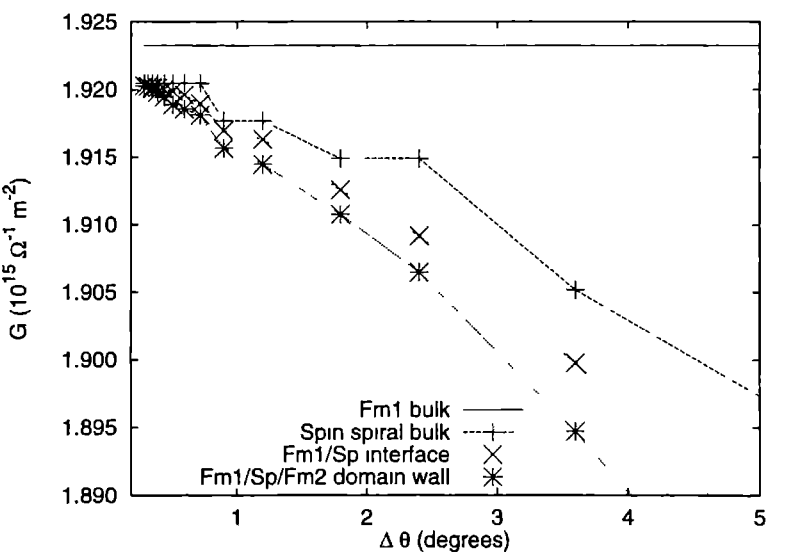


Figure 12.9: *Conductance of the Ni domain wall, in the small angle (i e large thickness) regime.*

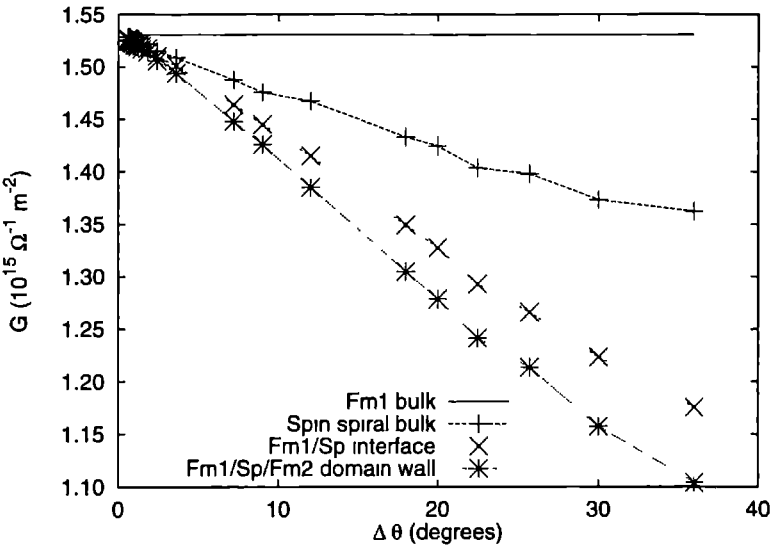


Figure 12.10: *Conductance of the Fe domain wall as a function of the angle $\Delta\theta$.*

quite large, because spin-up electrons suddenly find themselves in a spin-down environment, and vice-versa. But in the case of a very gradual change of the magnetisation, the spin of the electron has a chance to precess and therefore not find itself in an 'alien' environment and travel through the entire domain wall without scattering.

12.5.2 The iron domain wall

For the Fe domain wall we use bcc(100) layers. We follow the same procedure as for the Ni domain wall, again varying the thickness of the domain wall. Results are shown in figure 12.10 for the entire range of angles and in figure 12.11 for just the small angles. We see the same sort of behaviour as for Ni. At the thickness of the domain wall given in table 12.1, $\Delta\theta \approx 0\,65°$, the conductance through the domain wall is 99.6% of the bulk conductance.

12.5.3 The cobalt domain wall

In the case of the cobalt domain wall we use fcc(111) layers, because these resemble hcp, and because in the case of fcc(111) we have a single layer in the repeated stacking. Treating hcp cobalt still leads to computational problems, although it is in principle possible. The results for the conductances are shown in figures 12.12 and 12.13. The range of angles has been adapted to the much smaller thickness of the Co domain wall. The conductance for the domain wall at the experimental thickness, $\Delta\theta \approx 2.5°$ is about 99.5% of the bulk conductance.

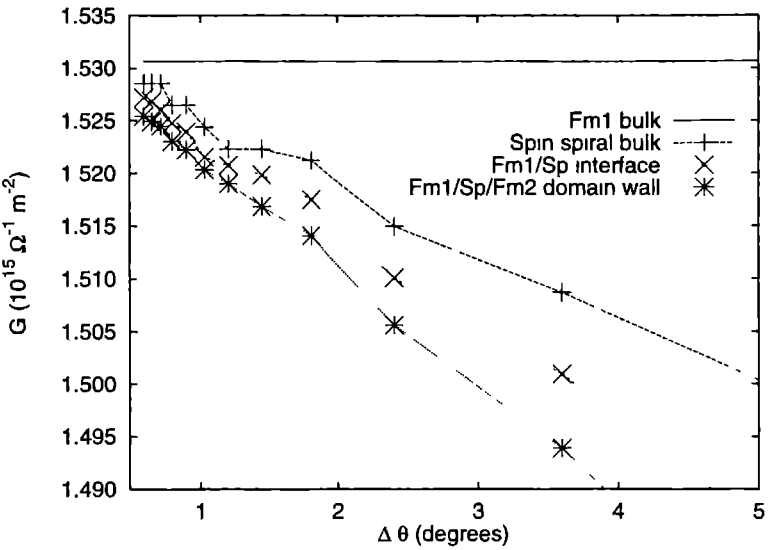


Figure 12 11: *Conductance of the Fe domain wall, for small angles*

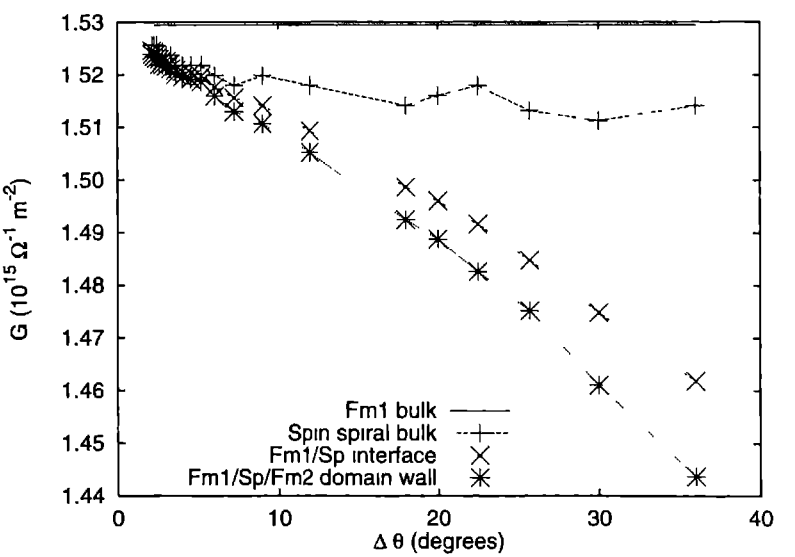


Figure 12.12: *Conductance of the Co domain wall as a function of the inter-subvolume angle* $\Delta\theta$

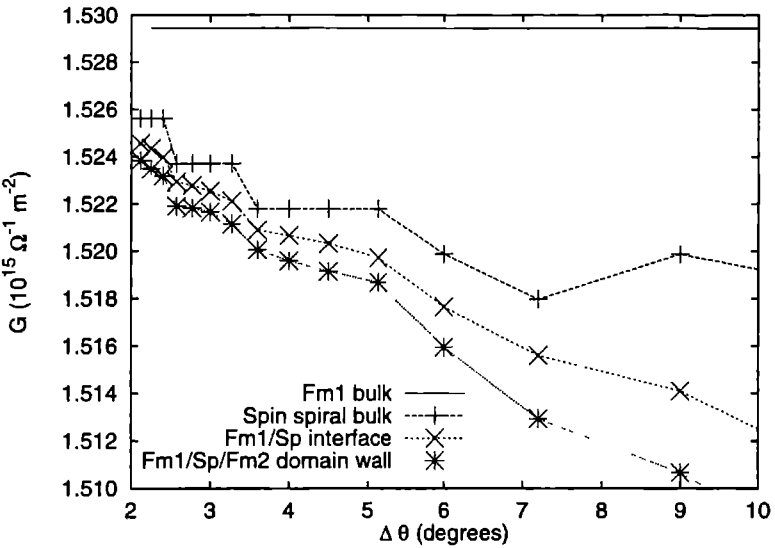


Figure 12.13: *Conductance of the Co domain wall, for small angles.*

Material	Conductance ($10^{15}\Omega^{-1}m^{-2}$)	% of bulk
Ni	1.92	99.85
Fe	1.52	99.6
Co	1.52	99.5

Table 12.2: *Conductances for the domain walls of experimental thickness.*

12.5.4 So do domain walls influence the conductance?

It seems that the conductance calculated for realistic domain walls of the experimental thickness, is more or less the same as for the bulk. The results for the three materials investigated are shown together in table 12.2. We see that the domain wall conductance is always 99.5% or more of the bulk value. This would indicate that the processes governing the conductance in a thick domain wall are really adiabatic as has been suggested in the literature[41]. In this thesis I shall not go into this problem in detail.

12.6 Abrupt domain walls

In this section I shall briefly give the conductances for the abrupt domain wall, as shown in the top part of figure 12.4. These conductances are calculated by adapting the method, by removing the spin-spiral part from figure 12.7. The values of the conductance for the different materials are shown in table 12.3. It is clear that the abrupt domain wall has a large influence on the conductance. This is because of the sudden change in environment for the electrons

Material	Conductance ($10^{15}\Omega^{-1}m^{-2}$)	% of bulk
Ni	0.80	41.6
Fe	0.44	29.0
Co	0.51	33.2

Table 12.3: *Conductances for the abrupt domain walls.*

travelling from the domain on the left to the domain on the right. In the next section we will see the abrupt domain wall again.

12.7 Non 180° domain walls

Up to now, we have only concentrated on domain walls where the two domains have their direction of magnetisation in exactly opposite directions, i.e. a total angle between the two ferromagnets of 180 degrees. In this section we shall examine the dependence of the domain wall conductance on this total angle. This has been done for both the domain walls of finite thickness and for the abrupt domain walls.

The results for the 'realistic' domain walls for all three materials are shown in figure 12.14. In this plot we show a sort of magneto-resistance (MR) value, namely the ratio of change of the conductance as function of the total angle. The wall in these calculations was 50 subvolumes thick for Ni and Fe and 25 subvolumes for Co. These thicknesses were chosen to get reasonably big results for the smaller angles. The behaviour in all three curves is more or less the same. We should also note that the effect keeps on increasing, which is logical, because an ever bigger total angle with a fixed number of layers means an ever greater angle between the layers. In figure 12.15 the results are shown for the abrupt domain walls as function of the angle between the magnetisation in the two domains. As we should expect, this curve has a period of 360 degrees.

12.8 Conclusions

In this chapter we have presented calculations for conductances in magnetic domain walls. These calculations were done both for abrupt and more realistic, thick domain walls.

The realistic domain walls were calculated using a frozen magnetisation configuration in the wall, with a constant angle $\Delta\theta$ between the subvolumes in the wall.

It turns out that the conductance for the realistic domain walls with experimental thicknesses is almost the same as the bulk conductance. This seems to be in agreement with the adiabatic picture mentioned in the literature[41].

However, experiments on iron whiskers[42, 43] have indicated a large magneto-resistance (MR) effect for the domain walls. This seems to be in contradiction to the results presented in this chapter. One of the reasons for this discrepancy may be the use of the constant $\Delta\theta$, but using a simple model for the reflection of an angle $\Delta\theta$, fitted to the data for the abrupt wall for small angles, makes this unlikely. Another reason may be that the realistic domain walls are thicker than the mean free path over which the electron retains its spin. This is a problem that needs some more study.

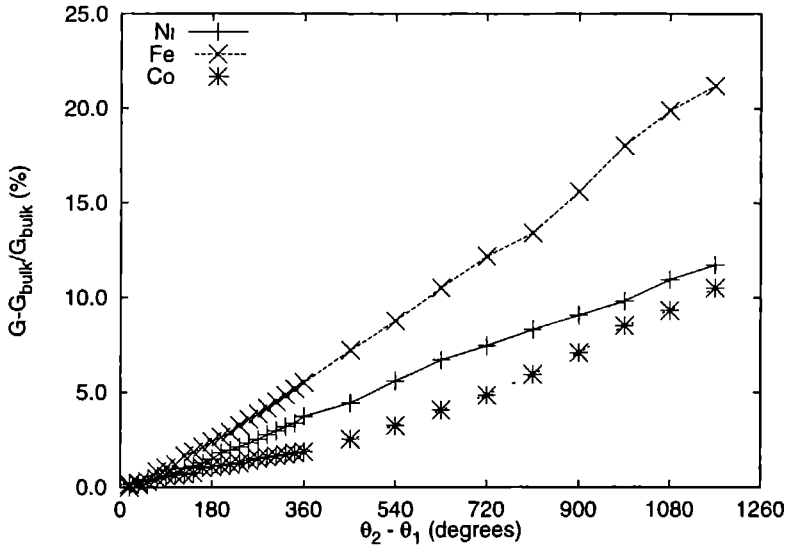


Figure 12.14: *Conductance of realistic domain walls for varying angles between the magnetisations in the two domains*

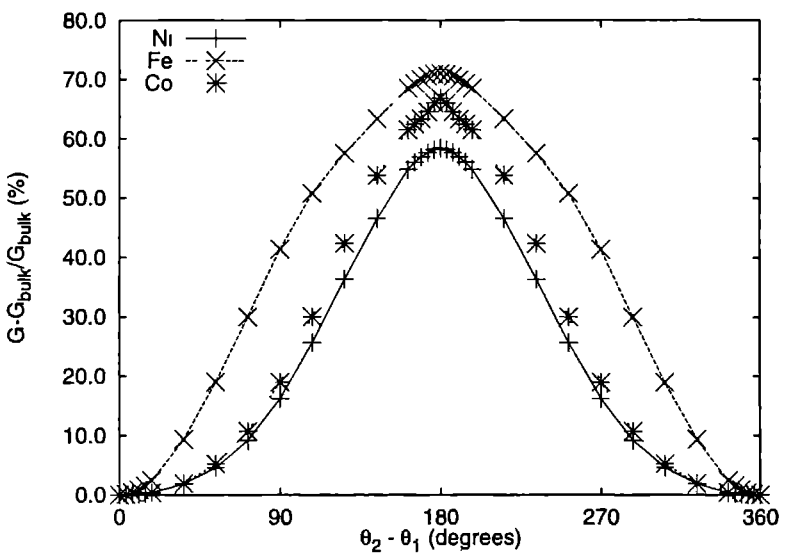


Figure 12.15: *Conductance of abrupt domain walls as a function of the angles between the magnetisations in the two domains.*

Chapter 13

The Cu/Co interface.... is there anything left to say?

Copper/Cobalt systems have been researched quite extensively in the last decade. Therefore the question in the title of this chapter is quite valid. It is my intention to discuss some results for a Cu/Co system that are new, simply because previously available methods were not able to perform these calculations.

In this chapter we shall discuss some of the conductance properties of a *single* interface between copper and cobalt. Previously multiple interfaces of Cu/Co in Cu-Co multilayers have been studied.

This chapter will be rather short, because I only want to mention those results which are new, and compare them to known results.

13.1 A description of the problem

In the past, Cu/Co multilayers have been used to study the Cu/Co interface and its conductance[44]. In that study multilayers of the type Cu_n/Co_n have been studied for n up to 8. For large n, these results should approach the situation for the single interface, which can be seen as the limiting case of $n \to \infty$. In a later section we shall investigate whether this assumption holds, by comparing the results for the single interface in the (100) direction with the data in [44].

13.1.1 Geometry

The situation under investigation is shown in figure 13.1. We have already seen a similar situation in figure 4.2 in section 4.4, where we described how to calculate conductances for such systems. As was described in section 4.4 we shall also use two subvolumes to perform the calculation. In the case of Co grown on Cu, both Cu and Co have a face-centered-cubic crystal structure. For our calculation we have chosen a lattice parameter of 6.769 a.u. (3.58 Å).

13.1.2 Convergence questions

Before we are able to handle the interface and its conductance, we have to converge muffin-tin bulk potentials. Because of the magnetic properties which are quite important in these calculations, a rather big $\mathbf{k}_{\parallel}$-point set was used.

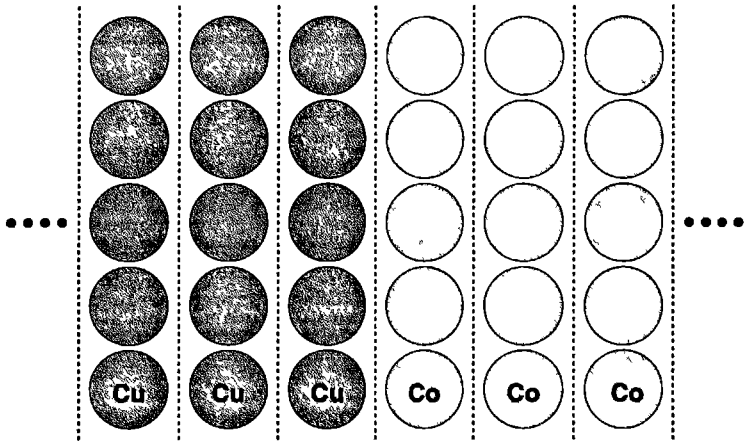


Figure 13.1: *Schematic picture of the Cu/Co interface.*

The Cu bulk is converged using 10 $\mathbf{k}_{\|}$-points in the irreducible part of the two-dimensional Brillouin zone (2D BZ) for the sum approximating the integral over the entire 2D BZ and using 32 energies in the energy integral. This is not an extremely big $\mathbf{k}_{\|}$-set, but going to the next bigger set does not change the physics.

For the Co bulk we use 22 $\mathbf{k}_{\|}$-points (in the irreducible part) to approximate the $\mathbf{k}_{\|}$-integral and 32 energies in the energy integral. This gives us a magnetic moment of 1.68 μB per atom, in good agreement with the experimental value of 1.72 μB (see [45]). This is more or less converged, since increasing the $\mathbf{k}_{\|}$-set doesn't significantly change the magnetic moment.

These bulk muffin-tin potentials are then used to construct the interface system. In other words we use bulk potentials up to the boundary between the two materials. This is not the 'real' situation, of course, but it is a good first approximation. We should really use self-consistent potentials for the interface case. In this chapter, however, I shall restrict myself to this non-self-consistent situation.

13.2 The Cu/Co (100) interface

The first interface we shall investigate is the interface where the layers are stacked in the (100) direction. This gives us a square unit cell in the two dimensions within the subvolume. In figure 13.2 the conductance for the interface is shown, as a function of the inverse of the number of $\mathbf{k}_{\|}$-points in the irreducible part of the 2D BZ, $N_{\mathbf{k}_{\|}}$. We use the inverse because we are eventually interested in the converged value in the limit of large $N_{\mathbf{k}_{\|}}$, which we find by extrapolating to $\frac{1}{N_{\mathbf{k}_{\|}}} = 0$. The values we end up with are $0.44 \times 10^{15}\Omega^{-1}m^{-2}$ for the majority spin (up) and $0.25 \times 10^{15}\Omega^{-1}m^{-2}$ for the minority spin (down). The error in both values is $\pm 0.01 \times 10^{15}\Omega^{-1}m^{-2}$. This means that the set of 465 $\mathbf{k}_{\|}$-points in the irreducible part is more or less converged (corresponding to 3600 $\mathbf{k}_{\|}$-points in the full 2D Brillouin zone).

To compare, in figure 13.3 we show the results presented in [44] for the Cu/Co (100)

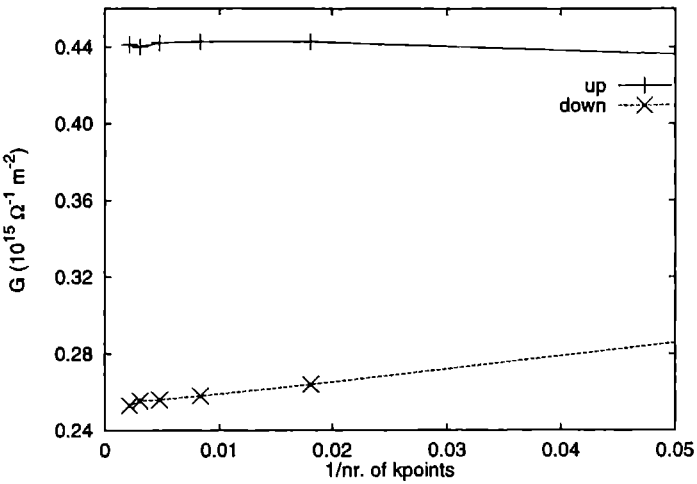


Figure 13.2: *Conductance for the Cu/Co (100) interface in both spin-channels as a function of the inverse of the number of* $\mathbf{k}_{\|}$*-points in the irreducible part of the 2D BZ.*

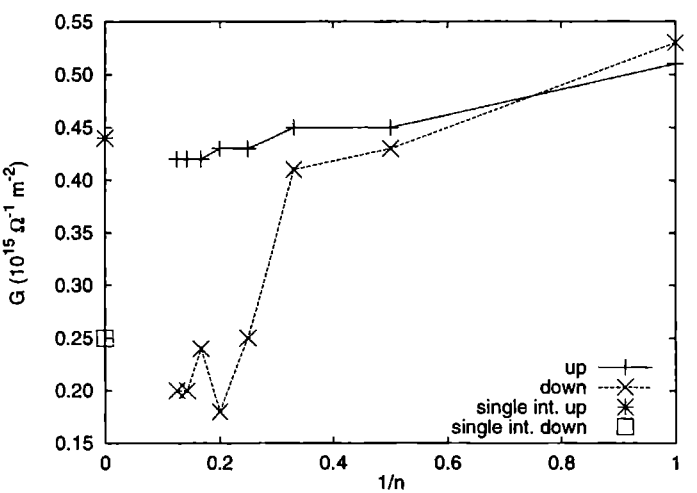


Figure 13.3: *Conductance for the* Cu_n/Co_n *bulk system, as a function of* $\frac{1}{n}$ *using data from [44], and the values for the single interface at* $\frac{1}{n} = 0$.

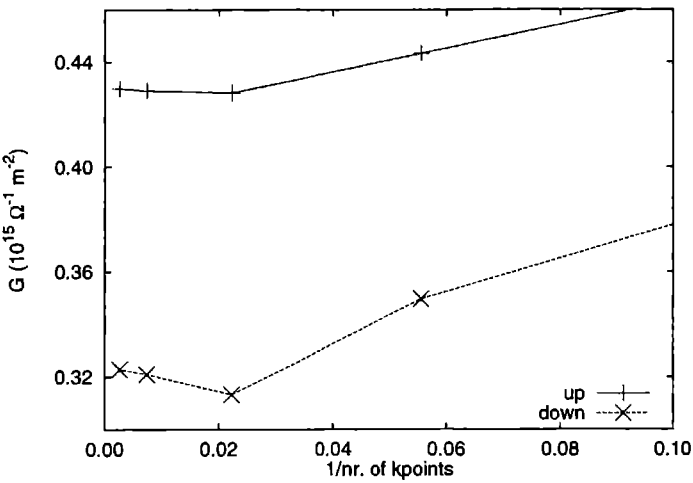


Figure 13.4: *Conductance for the Cu/Co (111) interface in each spin-channel as a function of the inverse of the number of* $\mathbf{k}_{\|}$*-points in the irreducible part of the 2D BZ.*

interface in a Cu_n/Co_n approximation, as a function of $\frac{1}{n}$. These results are slightly different from ours, but we have to take into account that the different results are obtained with different methods. The results in [44] are obtained using the linear muffin-tin orbital (LMTO) method. But more importantly the lattice parameter used in [44] is that of bulk fcc Cu, whereas in this chapter we use a slightly different lattice parameter, namely the average of the Cu and Co lattice parameter. Also because the Cu_n/Co_n system is constructed to be periodic, effects of constructive interference of states can lead to different results.

The advantage of our method is that it enables us to calculate the single interface, without having to approximate by calculating the Cu_n/Co_n bulk for large n. From figure 13.3 we can conclude that after using 16 layers in the unit-cell (8 layers of Cu and 8 layers of Co) the conductance is more or less converged, whereas our method will give the result for the single interface using only two layers in the calculation.

Another advantage is, that for implementing the Landauer formalism, we just calculate the band-structure at the Fermi energy, whereas using a method such as LMTO it is only possible to construct the entire band-structure. This means that using our method we avoid doing a lot of extra, unnecessary work.

Actually we can now use the transfer matrix for the single interface that we can construct, as a building block. For example, we can use the building block to investigate the resistance of a multilayer of Cu/Co, where the layers are really thick $n \gg 50$ and do not necessarily all have the same thickness[46].

13.3 The Cu/Co (111) interface

We have also examined the conductance for the single interface of Cu/Co with the subvolumes stacked in the fcc(111) direction. The convergence of the results is shown in figure 13.4. Using extrapolation we get the following values for the conductances: the majority conductance is $0.43 \pm 0.01 \times 10^{15}\Omega^{-1}m^{-2}$ and the minority conductance $0.32 \pm 0.01 \times 10^{15}\Omega^{-1}m^{-2}$.

The first thing to notice is that these values are slightly different from the ones for the (100) interface. This is logical, since the shape of the Fermi surfaces of both materials is different. Whereas for Cu the Fermi surface projected on the (100) Brillouin zone is more or less a circle with some extrusions where the necks are, the projection on the (111) BZ has a hole in the middle, because we are 'looking' straight down a neck.

We can compare these results with the values for Cu_3/Co_3, given in [47]. The conductance for the majority spin in that case is $0.41 \times 10^{15}\Omega^{-1}m^{-2}$ and for the minority spin $0.32 \times 10^{15}\Omega^{-1}m^{-2}$. These are close to the results that we find for the single interface.

13.4 Conclusions

In this chapter I have shown results for ab-initio conductance calculation for a single interface between Cu and Co. The results agree very well with other calculations for systems approaching the single interface. Future work on this subject could be the use of the transfer matrix building block to examine more elaborate multilayers of Cu and Co.

Chapter 14

Conductance in tunneling systems

In this chapter I shall show some results for conductance calculations in tunneling systems. The systems under investigation consist of surfaces separated by a distance d. The advantage of the embedding method in this case is that the entire tunneling system is handled as *one* quantum-system including the coupling to the two semi-infinite substrates. This is contrary to other methods, which calculate both sides of the system separately and then couple them later.

14.1 A description of the system

We shall examine systems which have a barrier through which we get tunneling behaviour. The barrier is the potential between two surfaces in close proximity. Since these systems retain their two-dimensional symmetry, we can use a layered description.

14.1.1 Geometry

The systems under investigation in this chapter all look as in figure 14.1. The region in which the calculation is performed is the stack of subvolumes between the two embedding planes. This region is built out of the two metal subvolumes on the sides, with one or more subvolumes of vacuum in between. The distance d is defined as the distance between the planes touching the tops of the muffin-tins in both surfaces. We have to remember this when comparing results for surfaces in different crystallographic directions, because the origin of d changes with each direction.

14.1.2 Method

The approach to the problem is to first perform a self-consistent calculation of a bulk configuration of the material under investigation. This gives us a self-consistent bulk muffin-tin potential, and the embedding potential and potential boundary conditions on the embedding planes. Then we put the bulk self-consistent muffin-tin potentials in the surface layers, construct a step potential in between and take the tunneling system to self-consistency. Once we have the converged tunneling potential, we calculate the conductance.

By approaching the problem like this, we make some approximations. A muffin-tin potential is not the best potential to use for the surface atoms, but we mainly want to do a feasibility study at this stage, to show that the method can handle systems incorporating a

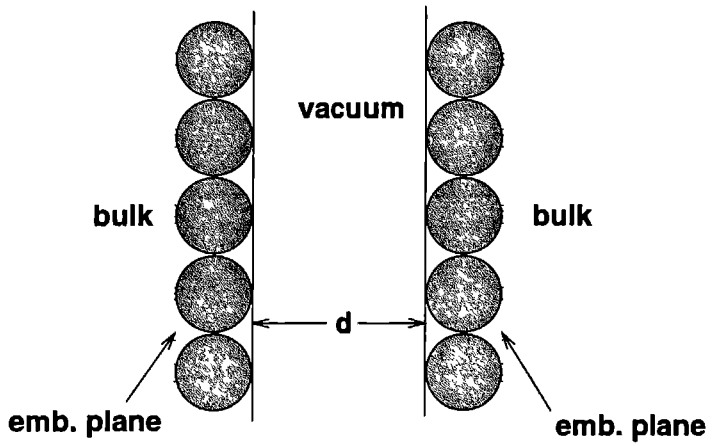


Figure 14.1: *Schematic picture of the tunneling system.*

tunneling barrier. Also the z-dependent potential in the vacuum is not the best choice, but again we just want to show that we are able to calculate systems like these.

14.2 Tunneling conductance from Al to Al

The first system we study is the system of two aluminium surfaces. This is a 'fast' system for a first test, because Al is an sp-material and we therefore only need to augment s and p states in the LAPW basis functions. We shall examine the tunneling conductance of two Al(100) surfaces as well as two Al(111) surfaces.

14.2.1 Al(100) surface to Al(100) surface

In figure 14.2 we show the tunnel conductances from an Al(100) surface to a Al(100) surface, as a function of the distance between the surfaces, d. We plot the logarithm of the conductance, to be able to see the behaviour for larger d, and also because the tunneling conductance should vary exponentially with d. Because the behaviour should be exponential, we have tried to fit an exponential function to the data points. This fit (also shown in figure 14.2) has the following form:

$$G = G_0 e^{-\alpha d} \tag{14.1}$$

We see that the exponential function is a perfect fit. From STM theory[48], we know that the tunneling current in STM for larger d behaves exponentially with an exponent:

$$-2\sqrt{\frac{2m\phi}{\hbar^2}}d \tag{14.2}$$

where ϕ is the workfunction of the metal. If we now take the exponent α that we get from the fit, and calculate a workfunction from it, we get $\phi = 5.34 \pm 0.09$ eV. This is actually bigger

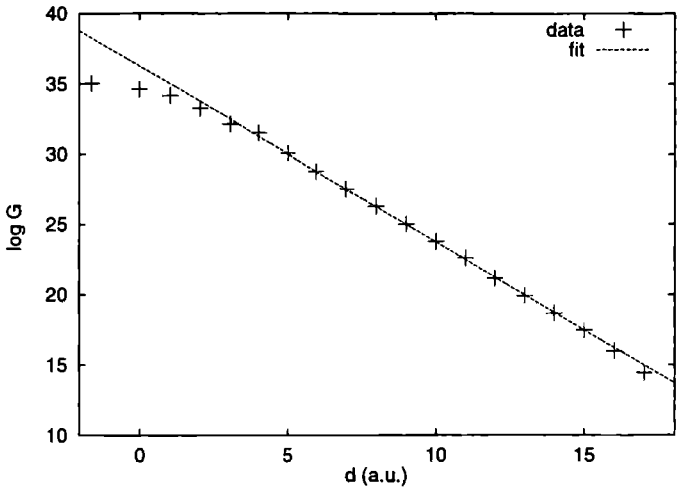


Figure 14.2: *Conductances calculated between two Al(100) surfaces. The straight line shows a fit with an exponential function.*

than the experimental value of the workfunction for the Al(100) surface (4.41 eV[32]). But we have to take into account that (14.2) arises from a theory which couples two separate quantum systems, and also that the workfunction which is calculated for the Al(100) surface using a muffin-tin potential within the embedding method is actually 5.6 eV (see section 11.1.2.1). So the agreement is actually much better.

One thing to notice is that for smaller distances d between the surfaces, the conductance does not lie on the exponential curve. This is the region where the surfaces are so close together that the simple model which gives rise to the exponential behaviour breaks down. For those distances it is essential to treat the entire system as one!

The one value for negative d is the value for the bulk conductance. The distance d is negative for the bulk, because of the definition of d as being the distance between planes touching the tops of the surface muffin-tins.

14.2.2 Al(111) surface to Al(111) surface

The tunnel conductances for the Al(111) surface to Al(111) surface is shown in figure 14.3. Again we have fitted the data with an exponential function. Using (14.2) this leads to a workfunction, $\phi = 6.20 \pm 0.05$ eV. The experimental value for the Al (111) surface is 4.24 eV[32]. The workfunction for a muffin-tin surface of Al(111) using our method is 5.15 eV. The discrepancy is even bigger, but still for the same reasons given for the (100) case.

There is again one negative d, different this time, because of the different stacking. Once again the data points for smaller d do not lie on the exponential fit.

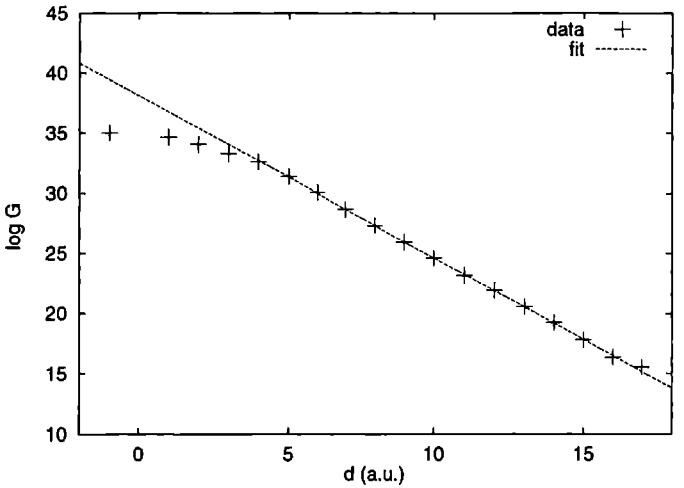


Figure 14.3: *Conductances calculated between two Al(111) surfaces. The straight line shows a fit with an exponential function.*

14.2.3 Comparing (100) with (111).

We can plot both conductance curves on the same figure, if we first align the distances d. We do this be defining a d', as the distance between the two surfaces, with $d' = 0$ at the bulk separation. This means that both plots shift with the d value of the bulk data point in figures 14.2 and 14.3. Both the data sets are shown in figure 14.4. We can clearly see the difference in slope of the fits in this plot, which shows the difference in workfunctions.

It seems that we can indeed handle tunneling systems with the embedding code and its conductance extension. Our results show an exponential dependance on the tunneling distance d, precisely as they should.

This was a test for a simple non-magnetic material. The next test we perform is for a metal which also has occupied d states, and is magnetic.

14.3 Tunneling conductance from Ni to Ni

Testing the method for the Ni(100) surface to Ni(100) surface tunneling, enables us to take a look at what happens in the case of spin-polarized tunneling. Since nickel is magnetic, and has a magnetic moment, we can try to tunnel both in a situation with the magnetisation for both surfaces pointing in the same direction (parallel) and in a situation with the magnetisation pointing in opposite directions (anti-parallel). We could even use the techniques described in chapter 12 to look at tunneling between two surfaces which have an arbitrary angle between the directions of their magnetisations, but that is beyond the scope of this thesis.

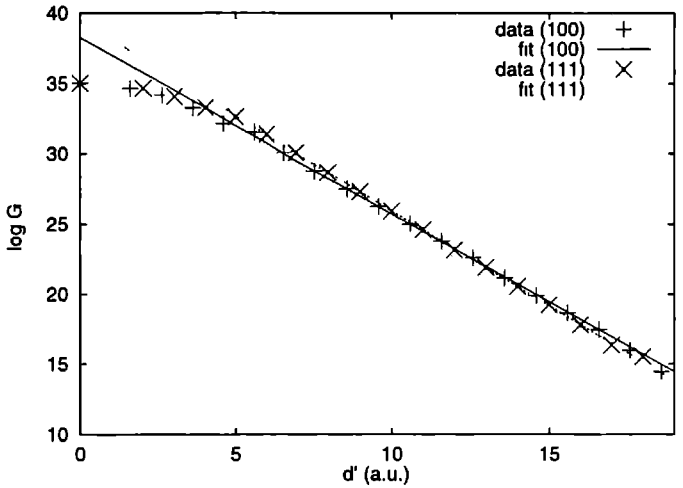


Figure 14.4: *Conductances for both Al surfaces as a function of d' ($d' = 0$ corresponds to bulk separation).*

14.3.1 Parallel configuration

The conductances for both spins are shown in figure 14.5. We notice that the conductance for each spin behaves like an exponential. We see that the up spins have a higher conductance. The exponents for both spins should be the same, because the workfunction is a spin-independent quantity. Because the fit was performed in the regime where the situation is not yet really asymptotic, we see a slight difference in the slopes for the up and down fit. From the fit we get a workfunction of 5.89 ± 0.13 eV.

14.3.2 Anti-parallel configuration

The results for the configuration where the magnetisations in the two surfaces point in opposite directions are shown in figure 14.6. These results are the same for both spins, as they should be. The slope of the fit yields a workfunction of 6.60 ± 0.08 eV. Let us now define the magneto-resistance effect as:

$$\mathrm{MR} = \frac{G_{\mathrm{P}\uparrow} + G_{\mathrm{P}\downarrow} - 2G_{\mathrm{AP}}}{2G_{\mathrm{AP}}} \tag{14.3}$$

where $G_{\mathrm{P}\uparrow}$ is the majority conductance in the parallel case, $G_{\mathrm{P}\downarrow}$ the minority conductance in the parallel case, and G_{AP} the conductance for one spin-direction in the anti-parallel case. Then, the MR effect for the tunnel barrier varies between 97% and 272% as we vary d.

14.4 Conclusions

In this chapter I have shown that the method is applicable to systems with a tunnel barrier. Applying this to the nickel system, we have shown that we can examine spin-polarised

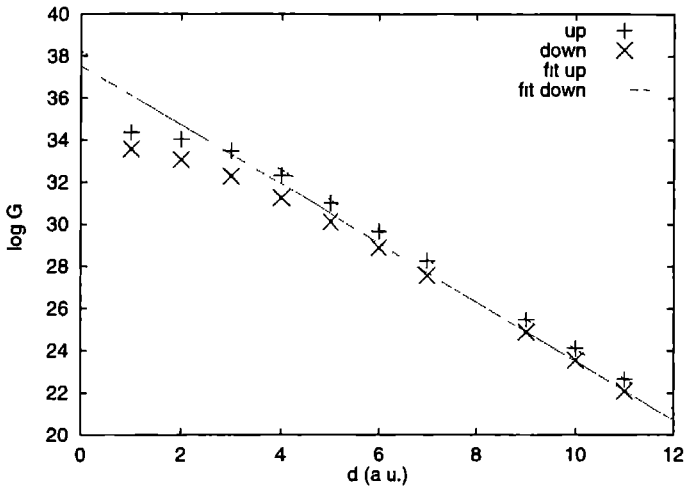


Figure 14.5: *Conductances for both spin directions in the Ni(100) surface to Ni(100) surface tunneling system. The magnetisations of the two surfaces are aligned parallel.*

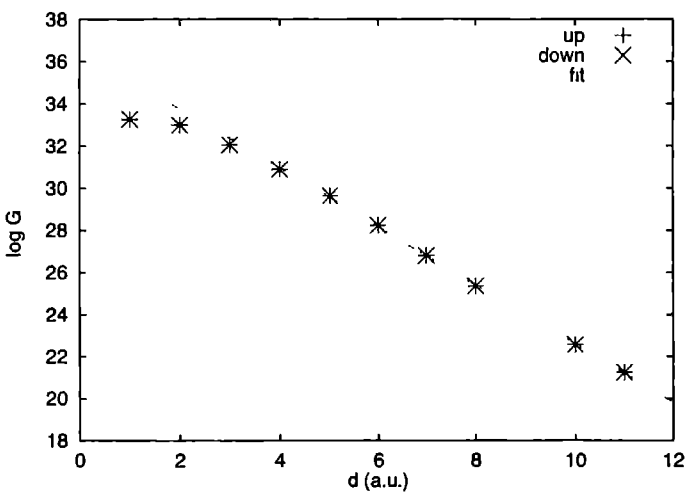


Figure 14.6: *Conductances for both spin directions in the Ni(100) surface to Ni(100) surface tunneling system. The magnetisations of the two surfaces are now aligned anti-parallel.*

tunneling in this way.

The major advantage of using this method to examine these tunneling systems is that we do not rely upon the density of states to guess the tunneling current, etc. We now know which states can conduct, and their transmission probabilities through the tunnel barrier.

The types of applications described in the last few chapters open up a large area of systems which can be investigated using the embedding method combined with the conductance formulation.

Part IV

Appendices

Appendix A

An electron's view of embedding

In this appendix I will try to explain what embedding is from an insider's point of view, and the obvious insider in electronic structure calculations is the electron.

A.1 The Land of Bulk

Let us try to get a feeling for what embedding does. We do this by performing a 'Gedanken'-experiment.

In analogy to Edwin A. Abbot's Flatland[49], we postulate *the Land of Bulk*. This is a two-dimensional world, like Flatland, but with a certain thickness. The Land of Bulk extends to infinity in both dimensions. The population of the Land of Bulk, consists of entities called *lectrons*. These lectrons live in spherical houses. Since the Land of Bulk is a very egalitarian world, all spheres have the same size and colour. Also the lectrons like order, therefore their houses are arranged in a very regular pattern. A top view of a part of the Land of Bulk will therefore look like figure A.1.

One day, one of the lectrons is visited by a scientist. The scientist asks the lectron: "Why are all your houses the same size and colour?". The lectron is confused, because it doesn't know anything but a world where all the houses are the same. Then the scientist asks: "What would happen to you and your people, if one of your houses was replaced by one of another size and colour?" Again the lectron is baffled and cannot help the scientist. The scientist then has the idea to perform an experiment. In his laboratory, he is going to reconstruct a world similar to the Land of Bulk, but with one house replaced by one of different size and colour. However the scientist soon realises that he does not have the resources to reconstruct the infinite reaches of the Land of Bulk. He could, of course, reconstruct only part of the Land of Bulk, but since the scientist is interested in how the change of one house will affect a population of lectrons, that is not an option. After talking some more with the lectron he is visiting, the scientist realises that these lectrons are quite gullible. That's when he starts to get the idea of tricking some lectrons into believing they are actually in the Land of Bulk with one different house.

But to be able to do that he has to know a lot more about the Land of Bulk. So he starts asking questions and in the process of a long conversation finds out the following things:

- lectrons do not have anything like telephone or radio, they only communicate 'face to face'.

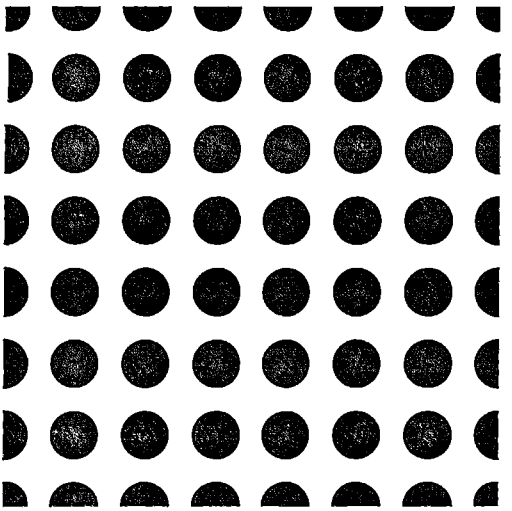


Figure A.1: *Top view of a piece of the* Land of Bulk, *where we can see that the houses are indeed all the same and arranged in a very regular pattern.*

- in the Land of Bulk there is an always-present fog, that limits the distance that the lectrons can 'see'.
- the distribution of lectrons in the Land of Bulk is rather uniform when averaged over an area of one house.
- by nature lectrons are lazy and not very curious.

The scientist leaves for his laboratory and does some preliminary research. He discovers a way to 'transport' lectrons from the Land of Bulk to his experiment without them noticing. Then he develops some cameras and special projection screens.

He then sets up the experiment as shown in figure A.2. The projection screens around the reconstructed part of the Land of Bulk are used to project live images from cameras mounted in the Land of Bulk, such that it looks from the inside as if outside of the screen there is more Land of Bulk. So if a lectron were to be transported into the simulation and would 'look' at the screen, it would not see a screen, but another part of the Land of Bulk, as it is used to.

Once the scientist has set all this up, and the cameras are rolling, he transports a part of the population out of the Land of Bulk into his simulation. The number of lectrons transported is about the number that would populate the same amount of area in the Land of Bulk. After waiting for a while, the scientist examines the experiment, and notices that the lectrons haven't even noticed they are no longer in the Land of Bulk. They move around as if nothing is wrong. Also the density of lectrons is the same as in the Land of Bulk. After taking some measurements, the scientist transports all the lectrons back to the Land of Bulk. Having learned that the lectrons are successfully fooled, the scientist thinks about the next stage of his experiment.

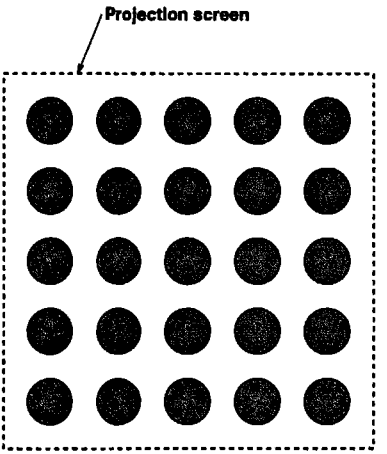


Figure A.2: *Top view of the reconstructed part of the Land of Bulk in the scientists laboratory.*

Since the scientist is really interested in the lectrons' reaction to a different house, he changes his reconstructed part of the Land of Bulk, to look as shown in figure A.3.

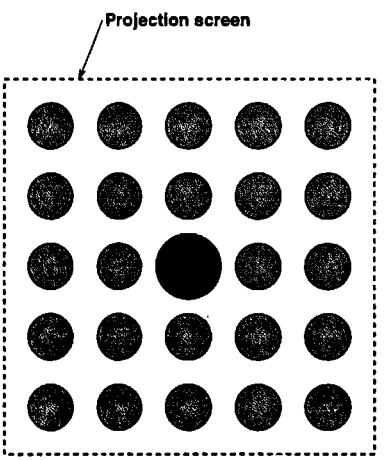


Figure A.3: *Top view of the reconstructed part of the Land of Bulk in the scientists laboratory, where the central house has been replaced by a bigger and darker one.*

Then the scientist again transports some lectrons into his simulation, the number of lectrons transported again being the same as the number of lectrons that would inhabit this area in the Land of Bulk. After letting the lectrons relax for a while, the scientist takes a

look at his simulation. What he sees surprises him. It turns out that the density of lectrons is no longer uniform. There are more lectrons around the center house in the simulation. So although the lectrons are not really curious, some of them took the trouble to go and investigate this strange house. The scientist also notices that the lectrons on the edge of the simulation, where the center house is not visible because of the fog, do not behave any different than in the previous simulation or in the Land of Bulk. After taking some measurements the lectrons are returned to the Land of Bulk.

The scientist is very happy to have found a way to investigate the interesting problem of the one different house in the Land of Bulk, without having to build an infinite copy of the Land of Bulk to do it. Also, the scientist realises that this technique with the cameras and the projection screens will enable him to perform many different experiments, changing the Land of Bulk locally.

Appendix B

Atomic solutions used in LAPW's and matrix elements

In this appendix I shall show how to solve the Schrödinger equation within the muffin-tins, to get the atomic solutions $u_{l\alpha}(r)$ and $\dot{u}_{l\alpha}(r)$ which are part of the linear augmented plane wave (LAPW). I shall also derive (partial) analytic expressions for some of the integrals involving these atomic solutions necessary to evaluate the Hamiltonian matrix elements.

B.1 Solution inside muffin-tins

The atomic solution inside the muffin-tin, $u_{l\alpha}(r)$, we take to be the large component of the solution of the Dirac equation in the limit of no spin-orbit coupling, following Koelling and Harmon[50]. We use a radial potential $V(r)$ (the spherically symmetric part of our full potential), and the solution of the Dirac equation is given by:

$$\Phi_{\kappa\mu} = \begin{pmatrix} g_\kappa \chi_{\kappa\mu} \\ -i f_\kappa \sigma_r \chi_{\kappa\mu} \end{pmatrix} \tag{B.1}$$

where g_κ is the large component we are interested in. $\Phi_{\kappa\mu}$ satisfies:

$$\frac{\mathrm{d}f_\kappa}{\mathrm{d}r} = \frac{1}{c}(V - E)g_\kappa + \left(\frac{\kappa - 1}{r}\right) f_\kappa \tag{B.2a}$$

$$\frac{\mathrm{d}g_\kappa}{\mathrm{d}r} = -\left(\frac{\kappa + 1}{r}\right) g_\kappa + 2Mc f_\kappa \tag{B.2b}$$

where E is the energy with the rest energy mc^2 subtracted, and:

$$M = m + \frac{1}{2c^2}(E - V) \tag{B.3a}$$

$$\kappa = \begin{cases} -(l+1) & \text{for } j = l + \frac{1}{2} \\ l & \text{for } j = l - \frac{1}{2} \end{cases} \tag{B.3b}$$

Equations (B.2) and (B.3) are in atomic units (c.f. [51]). If we differentiate the second equation again we get an equation for the large component g that can finally be written as:

$$-\frac{1}{2M}\left(\frac{\mathrm{d}^2 g}{\mathrm{d}r^2} + \frac{2}{r}\frac{\mathrm{d}g}{\mathrm{d}r} - \frac{l(l+1)}{r^2} g\right) + Vg - \frac{V'}{4M^2c^2}\frac{\mathrm{d}g}{\mathrm{d}r} - \left(\frac{\kappa + 1}{r}\right)\frac{V'}{4M^2c^2} g = Eg \tag{B.4}$$

The first line of (B.4) is the non-relativistic Schrödinger equation with M substituted for m to take care of mass-velocity effects. The first term of the second line is the Darwin term, and the second term of the second line is the spin-orbit term. We shall drop this spin-orbit term and notice that in that case (B.4) is equivalent to (6.21).

In the computer code we solve coupled equations derived from (B.2), to give us $\Phi_{\kappa\mu}$, from which we get the $u_{l\alpha}(r)$ by taking the large component. The coupled equations are solved using a predictor-corrector method, with a polynomial expansion as the starting point. The solution is generated on an exponential grid:

$$r_i = r_0(e^{(i-1)d_x} - 1) \tag{B.5}$$

with the first point zero and the last point on the muffin-tin sphere. The solution of the Dirac equation is calculated on a variable grid. The grid is variable in the sense that the number of grid points is doubled several times until the atomic solution has the desired accuracy.

Another set of coupled equations is solved in the same manner to give $\dot{u}_{l\alpha}(r)$. The boundary conditions on the two sets of coupled equations derived from (B.2) ensure that:

$$\int_\alpha \mathrm{d}r u_{l\alpha}(r)u_{l\alpha}(r) = 1 \tag{B.6a}$$

$$\int_\alpha \mathrm{d}r u_{l\alpha}(r)\dot{u}_{l\alpha}(r) = 0 \tag{B.6b}$$

Solving the Dirac equation is done in subroutine `ATOM_LAPW` and its subsidiaries. This means that after a call to subroutine `ATOM_LAPW`, with as input the spherically symmetric potential on the grid of (B.5), we get as output the solutions $u_{l\alpha}(r)$ and $\dot{u}_{l\alpha}(r)$ on the same grid.

B.2 Integrals involving $u_{l\alpha}(r)$ and $\dot{u}_{l\alpha}(r)$

The next pieces of information that we need in constructing Hamiltonian matrix elements using LAPW's are integrals using the atomic solutions. Some of these integrals are solved numerically, some are derived from other integrals.

First let us write down the Hamiltonian for which $u_{l\alpha}(r)$ and $\dot{u}_{l\alpha}(r)$ are the solutions. The Dirac equation is solved at a pivot energy $E_{l\alpha}$ for each angular quantum number l. The Hamiltonian at this energy is:

$$H(E_{l\alpha}) = -\frac{1}{2M(E_{l\alpha})}\left[\frac{1}{r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left(r^2\frac{\mathrm{d}}{\mathrm{d}r}\right) + \frac{l(l+1)}{r^2}\right] + V(r) - \frac{V'(r)}{4M(E_{l\alpha})^2c^2}\frac{\mathrm{d}}{\mathrm{d}r} \tag{B.7}$$

We are now interested in the relation between (6.21) and (B.7). We obtain:

$$\begin{aligned} H(E) &= H(E_{l\alpha}) + \frac{E_{l\alpha}}{2mc^2}H_0 - \frac{E}{2mc^2}H_0 + O(\frac{1}{c^4}) \\ &= H(E_{l\alpha}) + \frac{E_{l\alpha} - E}{2mc^2}H_0 + O(\frac{1}{c^4}) \end{aligned} \tag{B.8}$$

The equation solved in `ATOM_LAPW` is:

$$H(E_{l\alpha})u_{l\alpha} = E_{l\alpha}u_{l\alpha} \tag{B.9}$$

Differentiating this with respect to energy we get:

$$H(E_{l\alpha})\dot{u}_{l\alpha} + \dot{H}(E_{l\alpha})u_{l\alpha} = E_{l\alpha}\dot{u}_{l\alpha} + u_{l\alpha} \tag{B.10}$$

Multiplying with $r^2 u_{l\alpha}(r)$, integrating, and using (B.6) we obtain:

$$\left\langle u_{l\alpha}\middle|H(E_{l\alpha})\middle|\dot{u}_{l\alpha}\right\rangle + \left\langle u_{l\alpha}\middle|\dot{H}(E_{l\alpha})\middle|u_{l\alpha}\right\rangle = E_{l\alpha}\left\langle u_{l\alpha}\middle|\dot{u}_{l\alpha}\right\rangle + \left\langle u_{l\alpha}\middle|u_{l\alpha}\right\rangle \tag{B.11}$$

or

$$\left\langle u_{l\alpha}\middle|H(E_{l\alpha})\middle|\dot{u}_{l\alpha}\right\rangle = 1 - \left\langle u_{l\alpha}\middle|\dot{H}(E_{l\alpha})\middle|u_{l\alpha}\right\rangle \tag{B.12}$$

where $\dot{H}(E_{l\alpha})$ is given by:

$$\dot{H}(E_{l\alpha}) = -\frac{H(E_{l\alpha})}{2mc^2} + \frac{V(r)}{2mc^2} + O(\frac{1}{c^4}) \tag{B.13}$$

From (B.13) and using (B.9) and (B.10) we get:

$$\left\langle u_{l\alpha}\middle|\dot{H}(E_{l\alpha})\middle|u_{l\alpha}\right\rangle = -\frac{E_{l\alpha}}{2mc^2} + \frac{1}{2mc^2}\left\langle u_{l\alpha}\middle|V\middle|u_{l\alpha}\right\rangle \tag{B.14}$$

And we obtain:

$$\left\langle u_{l\alpha}\middle|H(E_{l\alpha})\middle|\dot{u}_{l\alpha}\right\rangle = 1 + \frac{E_{l\alpha}}{2mc^2} - \frac{1}{2mc^2}\left\langle u_{l\alpha}\middle|V\middle|u_{l\alpha}\right\rangle \tag{B.15}$$

When we multiply (B.10) with $r^2\dot{u}_{l\alpha}(r)$ and integrate we get:

$$\left\langle \dot{u}_{l\alpha}\middle|H(E_{l\alpha})\middle|\dot{u}_{l\alpha}\right\rangle = -\left\langle \dot{u}_{l\alpha}\middle|\dot{H}(E_{l\alpha})\middle|u_{l\alpha}\right\rangle + E_{l\alpha}\left\langle \dot{u}_{l\alpha}\middle|\dot{u}_{l\alpha}\right\rangle + \left\langle \dot{u}_{l\alpha}\middle|u_{l\alpha}\right\rangle \tag{B.16}$$

The last term is zero ((B.6b)) and substituting (B.13), we obtain:

$$\left\langle \dot{u}_{l\alpha}\middle|H(E_{l\alpha})\middle|\dot{u}_{l\alpha}\right\rangle = \frac{1}{2mc^2}\left\langle \dot{u}_{l\alpha}\middle|H(E_{l\alpha})\middle|u_{l\alpha}\right\rangle - \frac{1}{2mc^2}\left\langle \dot{u}_{l\alpha}\middle|V(r)\middle|u_{l\alpha}\right\rangle + E_{l\alpha}\left\langle \dot{u}_{l\alpha}\middle|\dot{u}_{l\alpha}\right\rangle \tag{B.17}$$

The first term is zero, because $H(E_{l\alpha})\middle|u_{l\alpha}\right\rangle = E_{l\alpha}\middle|u_{l\alpha}\right\rangle$ and $\left\langle \dot{u}_{l\alpha}\middle|E_{l\alpha}\middle|u_{l\alpha}\right\rangle = 0$.
To recap, we now have derived the following integrals:

$$\left\langle u_{l\alpha}\middle|H(E_{l\alpha})\middle|u_{l\alpha}\right\rangle = E_{l\alpha} \tag{B.18a}$$

$$\left\langle \dot{u}_{l\alpha}\middle|H(E_{l\alpha})\middle|u_{l\alpha}\right\rangle = 0 \tag{B.18b}$$

$$\left\langle u_{l\alpha}\middle|H(E_{l\alpha})\middle|\dot{u}_{l\alpha}\right\rangle = 1 + \frac{E_{l\alpha}}{2mc^2} - \frac{1}{2mc^2}\left\langle u_{l\alpha}\middle|V\middle|u_{l\alpha}\right\rangle \tag{B.18c}$$

$$\left\langle \dot{u}_{l\alpha}\middle|H(E_{l\alpha})\middle|\dot{u}_{l\alpha}\right\rangle = -\frac{1}{2mc^2}\left\langle \dot{u}_{l\alpha}\middle|V\middle|u_{l\alpha}\right\rangle + E_{l\alpha}\left\langle \dot{u}_{l\alpha}\middle|\dot{u}_{l\alpha}\right\rangle \tag{B.18d}$$

Using the integrals from (B.18) we now look at the integrals containing atomic solutions in (6.26). These 4 integrals are treated in the next 4 sections.

B.2.1 $\int_\alpha r^2 \mathrm{d}r\ u_{l,\alpha}(r) H(E) u_{l,\alpha}(r)$

For this first integral, we straightforwardly write out $H(E)$ using (B.8) to obtain:

$$\begin{aligned}\int_\alpha r^2\mathrm{d}r\ u_{l,\alpha}(r)H(E)u_{l,\alpha}(r) &= \int_\alpha r^2\mathrm{d}r\ u_{l,\alpha}(r)\left[H(E_{l\alpha}) + \frac{E_{l\alpha}-E}{2mc^2}H_0\right]u_{l,\alpha}(r)\\ &= \int_\alpha r^2\mathrm{d}r\ u_{l,\alpha}(r)E_{l\alpha}u_{l,\alpha}(r) +\\ &\quad \int_\alpha r^2\mathrm{d}r\ u_{l,\alpha}(r)\frac{E_{l\alpha}-E}{2mc^2}H_0u_{l,\alpha}(r)\\ &= E_{l\alpha}\left\langle u_{l,\alpha}\middle|u_{l,\alpha}\right\rangle + \frac{E_{l\alpha}-E}{2mc^2}\left\langle u_{l,\alpha}\middle|H_0\middle|u_{l,\alpha}\right\rangle \end{aligned} \tag{B.19}$$

The integral $\left\langle u_{l,\alpha}\middle|H_0\middle|u_{l,\alpha}\right\rangle$ we will discuss later.

B.2.2 $\int_\alpha r^2 \mathrm{d}r u_{l,\alpha}(r) H(E) \dot{u}_{l,\alpha}(r)$

This integral is a bit more tricky. We have to use several substitutions. First we substitute (B.8) and obtain:

$$\begin{aligned}\int_\alpha r^2\mathrm{d}r\ u_{l,\alpha}(r)H(E)\dot{u}_{l,\alpha}(r) &= \int_\alpha r^2\mathrm{d}r\ u_{l,\alpha}(r)\left(H(E_{l\alpha}) + \frac{E_{l\alpha}-E}{2mc^2}H_0\right)\dot{u}_{l,\alpha}(r)\\ &= \int_\alpha r^2\mathrm{d}r\ u_{l,\alpha}(r)H(E_{l\alpha})\dot{u}_{l,\alpha}(r)\\ &\quad + \int_\alpha r^2\mathrm{d}r\ u_{l,\alpha}(r)\frac{E_{l\alpha}-E}{2mc^2}H_0\dot{u}_{l,\alpha}(r)\end{aligned}$$

Then we substitute (B.14):

$$\begin{aligned}\int_\alpha r^2\mathrm{d}r\ u_{l,\alpha}(r)H(E)\dot{u}_{l,\alpha}(r) = &-\int_\alpha r^2\mathrm{d}r\ u_{l,\alpha}(r)\dot{H}(E_{l\alpha})u_{l,\alpha}(r)\\ &+ E_{l\alpha}\int_\alpha r^2\mathrm{d}r\ u_{l,\alpha}(r)\dot{u}_{l,\alpha}(r) + \int_\alpha r^2\mathrm{d}r\ u_{l,\alpha}(r)u_{l,\alpha}(r)\\ &+ \int_\alpha r^2\mathrm{d}r\ u_{l,\alpha}(r)\frac{E_{l\alpha}-E}{2mc^2}H_0\dot{u}_{l,\alpha}(r)\end{aligned}$$

and use (B.13):

$$\begin{aligned}\int_\alpha r^2\mathrm{d}r\ u_{l,\alpha}(r)H(E)\dot{u}_{l,\alpha}(r) =& \frac{1}{2mc^2}\int_\alpha r^2\mathrm{d}r\ u_{l,\alpha}(r)\Big(H(E_{l\alpha})-V(r)\Big)u_{l,\alpha}(r)\\ &+ E_{l\alpha}\left\langle u_{l,\alpha}\middle|\dot{u}_{l,\alpha}\right\rangle + \left\langle u_{l,\alpha}\middle|u_{l,\alpha}\right\rangle + \frac{E_{l\alpha}-E}{2mc^2}\left\langle u_{l,\alpha}\middle|H_0\middle|\dot{u}_{l,\alpha}\right\rangle\\ =& 1 + \frac{E_{l\alpha}}{2mc^2} - \frac{1}{2mc^2}\left\langle u_{l\alpha}\middle|V\middle|u_{l,\alpha}\right\rangle\\ &+ \frac{E_{l\alpha}-E}{2mc^2}\left\langle u_{l,\alpha}\middle|H_0\middle|\dot{u}_{l,\alpha}\right\rangle\end{aligned} \tag{B.20}$$

We save the terms $\left\langle u_{l,\alpha}\middle|V\middle|u_{l,\alpha}\right\rangle$ and $\left\langle u_{l,\alpha}\middle|H_0\middle|\dot{u}_{l,\alpha}\right\rangle$ for later.

B.2.3 $\int_\alpha r^2 \mathrm{d}r \dot{u}_{l,\alpha}(r) H(E) u_{l,\alpha}(r)$

This integral is rather straightforward, we just substitute (B.8) to obtain:

$$\begin{aligned}\int_\alpha r^2\mathrm{d}r\ \dot{u}_{l,\alpha}(r)H(E)u_{l,\alpha}(r) &= \int_\alpha r^2\mathrm{d}r\ \dot{u}_{l,\alpha}(r)\left(H(E_{l\alpha}) + \frac{E_{l\alpha}-E}{2mc^2}H_0\right)u_{l,\alpha}(r)\\ &= E_{l\alpha}\left\langle \dot{u}_{l,\alpha}\middle|u_{l,\alpha}\right\rangle + \frac{E_{l\alpha}-E}{2mc^2}\left\langle \dot{u}_{l,\alpha}\middle|H_0\middle|u_{l,\alpha}\right\rangle\\ &= \frac{E_{l\alpha}-E}{2mc^2}\left\langle \dot{u}_{l,\alpha}\middle|H_0\middle|u_{l,\alpha}\right\rangle \end{aligned} \tag{B.21}$$

Here we have another integral to be discussed later, namely $\left\langle \dot{u}_{l,\alpha}\middle|H_0\middle|u_{l,\alpha}\right\rangle$.

B.2.4 $\int_\alpha r^2 \mathrm{d}r \dot{u}_{l,\alpha}(r) H(E) \dot{u}_{l,\alpha}(r)$

This last integral containing atomic solution in (6.26) needs several substitutions again. First substitute (B.8):

$$\int_\alpha r^2\mathrm{d}r\ \dot{u}_{l,\alpha}(r)H(E)\dot{u}_{l,\alpha}(r) = \int_\alpha r^2\mathrm{d}r\ \dot{u}_{l,\alpha}(r)\left(H(E_{l\alpha}) + \frac{E_{l\alpha}-E}{2mc^2}H_0\right)\dot{u}_{l,\alpha}(r)$$

then we substitute (B.15):

$$\begin{aligned}\int_\alpha r^2\mathrm{d}r\ \dot{u}_{l,\alpha}(r)H(E)\dot{u}_{l,\alpha}(r) &= \int_\alpha r^2\mathrm{d}r\ \dot{u}_{l,\alpha}(r)\left(-\dot{H}(E_{l\alpha})u_{l,\alpha} + E_{l\alpha}\dot{u}_{l,\alpha} + u_{l,\alpha}(r)\right)\\ &\quad + \frac{E_{l\alpha}-E}{2mc^2}\int_\alpha r^2\mathrm{d}r\ \dot{u}_{l,\alpha}H_0\dot{u}_{l,\alpha}\end{aligned}$$

and finally (B.13):

$$\begin{aligned}\int_\alpha r^2\mathrm{d}r\ \dot{u}_{l,\alpha}(r)H(E)\dot{u}_{l,\alpha}(r) &= \int_\alpha r^2\mathrm{d}r\ \dot{u}_{l,\alpha}(r)\left(H(E_{l\alpha})-V(r)\right)u_{l,\alpha}(r) + E_{l\alpha}\left\langle \dot{u}_{l,\alpha}\middle|\dot{u}_{l,\alpha}\right\rangle\\ &\quad + \frac{E_{l\alpha}-E}{2mc^2}\left\langle \dot{u}_{l,\alpha}\middle|H_0\middle|\dot{u}_{l,\alpha}\right\rangle\\ &= -\frac{1}{2mc^2}\left\langle \dot{u}_{l,\alpha}\middle|V\middle|u_{l,\alpha}\right\rangle + E_{l\alpha}\left\langle \dot{u}_{l,\alpha}\middle|\dot{u}_{l,\alpha}\right\rangle\\ &\quad + \frac{E_{l\alpha}-E}{2mc^2}\left\langle \dot{u}_{l,\alpha}\middle|H_0\middle|\dot{u}_{l,\alpha}\right\rangle\end{aligned} \tag{B.22}$$

The integrals in (B.22) will be discussed in the next section.

B.3 Which integrals have to be solved numerically?

In the previous section we have written the integrals that are needed to calculate the Hamiltonian in terms of other integrals involving the atomic solutions. Some of these integrals have to be solved numerically. These are:

$$\begin{gathered}\left\langle \dot{u}_{l,\alpha}\middle|\dot{u}_{l,\alpha}\right\rangle\\ \left\langle u_{l,\alpha}\middle|V\middle|u_{l,\alpha}\right\rangle\\ \left\langle \dot{u}_{l,\alpha}\middle|V\middle|u_{l,\alpha}\right\rangle\end{gathered} \tag{B.23}$$

The other integrals involving H_0 can be further rewritten. First let us look at $\left\langle u_{l,\alpha}\middle|H_0\middle|u_{l,\alpha}\right\rangle$. Using the expression for H_0 as defined in (6.23), we can write:

$$\begin{aligned}
\left\langle u_{l,\alpha}\middle|H_0\middle|u_{l,\alpha}\right\rangle &= \int_\alpha r^2 \mathrm{d}r u_{l,\alpha}(r)\left[-\frac{1}{2m}\frac{1}{r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left(r^2\frac{\mathrm{d}}{\mathrm{d}r}\right)+\frac{l(l+1)}{2mr^2}\right]u_{l,\alpha}(r)\\
&= -\frac{1}{2m}\int_\alpha r^2 \mathrm{d}r u_{l,\alpha}(r)\frac{\mathrm{d}}{\mathrm{d}r}\left(r^2 u'_{l,\alpha}(r)\right)\\
&\quad + \frac{l(l+1)}{2m}\int_\alpha \mathrm{d}r u_{l,\alpha}(r)u_{l,\alpha}(r)\\
&= -\frac{1}{2m}\left[r^2 u'_{l,\alpha}(r)u_{l,\alpha}(r)\right]_0^{r^\alpha_{MT}} + \frac{1}{2m}\int_\alpha r^2 \mathrm{d}r u'_{l,\alpha}(r)u'_{l,\alpha}(r)\\
&\quad + \frac{l(l+1)}{2m}\int_\alpha \mathrm{d}r u_{l,\alpha}(r)u_{l,\alpha}(r)\\
&= -\frac{1}{2m}\left(r^\alpha_{MT}\right)^2 u_{l,\alpha}(r^\alpha_{MT})u'_{l,\alpha}(r^\alpha_{MT}) + \frac{1}{2m}\left\langle u'_{l,\alpha}\middle|u'_{l,\alpha}\right\rangle\\
&\quad + \frac{l(l+1)}{2m}\int_\alpha \mathrm{d}r \left(u_{l,\alpha}(r)\right)^2
\end{aligned} \tag{B.24}$$

We now have two new integrals to calculate numerically. Note that the last integral in (B.24) does not contain a factor r^2. The radial derivative of the atomic solution, $u'_{l,\alpha}(r)$, is a by-product of solving for $u_{l,\alpha}(r)$, so we can actually calculate (B.24).

The derivation of the other three integrals containing H_0 goes along similar lines. I shall give them without derivation.

$$\begin{aligned}
\left\langle \dot{u}_{l,\alpha}\middle|H_0\middle|u_{l,\alpha}\right\rangle &= -\frac{1}{2m}\left(r^\alpha_{MT}\right)^2 \dot{u}_{l,\alpha}(r^\alpha_{MT})u'_{l,\alpha}(r^\alpha_{MT}) + \frac{1}{2m}\left\langle \dot{u}'_{l,\alpha}\middle|u'_{l,\alpha}\right\rangle\\
&\quad + \frac{l(l+1)}{2m}\int_\alpha \mathrm{d}r \dot{u}_{l,\alpha}(r)u_{l,\alpha}(r)
\end{aligned} \tag{B.25}$$

$$\begin{aligned}
\left\langle u_{l,\alpha}\middle|H_0\middle|\dot{u}_{l,\alpha}\right\rangle &= -\frac{1}{2m}\left(r^\alpha_{MT}\right)^2 u_{l,\alpha}(r^\alpha_{MT})\dot{u}'_{l,\alpha}(r^\alpha_{MT}) + \frac{1}{2m}\left\langle u'_{l,\alpha}\middle|\dot{u}'_{l,\alpha}\right\rangle\\
&\quad + \frac{l(l+1)}{2m}\int_\alpha \mathrm{d}r u_{l,\alpha}(r)\dot{u}_{l,\alpha}(r)
\end{aligned} \tag{B.26}$$

and

$$\begin{aligned}
\left\langle \dot{u}_{l,\alpha}\middle|H_0\middle|\dot{u}_{l,\alpha}\right\rangle &= -\frac{1}{2m}\left(r^\alpha_{MT}\right)^2 \dot{u}_{l,\alpha}(r^\alpha_{MT})\dot{u}'_{l,\alpha}(r^\alpha_{MT}) + \frac{1}{2m}\left\langle \dot{u}'_{l,\alpha}\middle|\dot{u}'_{l,\alpha}\right\rangle\\
&\quad + \frac{l(l+1)}{2m}\int_\alpha \mathrm{d}r \dot{u}_{l,\alpha}(r)\dot{u}_{l,\alpha}(r)
\end{aligned} \tag{B.27}$$

In the last section of this appendix I shall list the outputs of subroutine `ATOM_LAPW`.

B.4 Quantities calculated in `ATOM_LAPW`

To summarize the output of `ATOM_LAPW`, I shall list the most important outputs:

1. The atomic solutions, $u_{l,\alpha}(r)$ and $\dot{u}_{l,\alpha}(r)$, on a grid inside the muffin-tin (needed for the augmentation part of the LAPW).

2. As an automatic extra, the radial derivatives of both solutions, $u'_{l,\alpha}(r)$ and $\dot{u}'_{l,\alpha}(r)$, on a grid inside the muffin-tin.

3. The values of all four functions on the muffin-tin radius r^{α}_{MT}.

4. The normalisation integral: $\left\langle \dot{u}_{l,\alpha} \middle| \dot{u}_{l,\alpha} \right\rangle$.

5. $\left\langle u_{l,\alpha} \middle| H(E_{l\alpha}) \middle| \dot{u}_{l,\alpha} \right\rangle$. To calculate this integral, the expectation value of the potential $V(r)$, $\left\langle u_{l,\alpha} \middle| V \middle| u_{l,\alpha} \right\rangle$, is calculated numerically and put into (B.18c).

6. $\left\langle \dot{u}_{l,\alpha} \middle| H(E_{l\alpha}) \middle| \dot{u}_{l,\alpha} \right\rangle$. To calculate this integral, the following integral of the potential $V(r)$, $\left\langle \dot{u}_{l,\alpha} \middle| V \middle| u_{l,\alpha} \right\rangle$, is calculated numerically and put into (B.18d)

7. The expectation value of the potential and the other integral over the potential, mentioned in the previous two points are output as well.

8. The four integrals in (B.24), (B.25), (B.26), and (B.27) are calculated, by calculating the integrals in the expressions numerically.

These are all the quantities needed in subroutine MTME_LAPW, the subroutine that calculates the Muffin-Tin Matrix Elements for the LAPW basis.

Summary

One of the problems that arise when performing electronic structure calculations in solids is the fact that sometimes the symmetry of the system can break down locally. A good example of this is a surface. At a surface the translational symmetry of the system is disturbed in one direction, namely perpendicular to the surface. This thesis describes a method that is able to *handle locally broken symmetries* by *embedding* the system with the broken symmetry in a system with unbroken symmetry.

The process of developing a new method can chronologically be partitioned into three steps. Logically, the first logical step is to construct and derive the method mathematically. The next step is to implement this mathematical construct into a computer code. Once we have a computer code we eventually use this code to perform calculations.

Before concentrating on the new method which solves only a small part of the problem of electronic structure calculations, it is good to examine some of the very basic problems involved in these calculations. This is the subject of the introduction in chapter 1. In this chapter it is explained how we approximate the Schrödinger equation of the crystal with about 10^{24} interacting particles by a one-electron Schrödinger equation known as the Kohn-Sham equation. The new method for solving the symmetry-breaking problem is also compared to the more traditional method for dealing with this problem, which is also briefly discussed in chapter 1.

Part One

In the first part of this thesis, we concentrate on the theory of the method. In chapter 2 the general method of embedding is discussed. This is a description of the method that is not based on a certain geometry. The first half of chapter 2 together with appendix A explains the idea behind embedding. As the name already suggests, the method really does *embed* the part of the system with the broken symmetry in a system with intact symmetry. The second half of chapter 2 gives the mathematical framework, linking the embedding potential to the Green function. The embedding equation that is introduced in section 2.3 is the basis of this entire thesis.

Chapter 3 describes the embedding method for a set of systems with a certain geometry, namely those that can be constructed from layers. This is the description of choice for such systems as surfaces and interfaces. In this chapter the link between embedding theory and scattering theory is made. Originally this is examined only as a way to calculate the embedding potential more efficiently, but as we show in chapter 4, the scattering properties can be used to perform some very interesting calculations on conductance problems. The introduction of the scattering properties in the embedding method was a major factor in making the method much more efficient, but also more interesting for certain types of problems, e.g. the

conductance problem.

In chapter 4 the conductance angle is worked out in detail. Using the Landauer formalism it is shown how the scattering properties can be used to calculate conductances through stacks of layers (i.e. interfaces). As is shown in this chapter some interesting new problems in describing the system in terms of basis functions arise when looking at electron transport.

Part Two

The second part concentrates on the implementation of the mathematics into a computer code. Using the programming language FORTRAN (FORmula TRANslation) this could be a rather straightforward process, but as is described in part 2 of this thesis it is not. In chapter 5 the logical structure of the method is discussed and is broken up into three logical modules: `GROW`, `FISH`, and `MIX`.

The `GROW` module of the method is implemented in chapter 6. This module solves the Schrödinger equation using a potential as input and eventually giving a charge-density as output. In this chapter the choice of basis is discussed, followed by the derivation of the Hamiltonian matrix elements in this basis (see also appendix B). Once the Hamiltonian matrix has given us the Green function matrix, we can get at properties of interest as the density of states and the charge-density. Instead of listing and describing all details of the implementation, some key points are examined in detail and at the end of the chapter an overview of the structure of the `GROW` module is given.

In chapter 7 we examine the `FISH` module of the code. This module takes a charge-density as input and uses this to solve Poisson's equation to generate a potential. In this chapter we encounter some problems because of our choice in chapter 3 to use a layered geometry with so-called *subvolumes*. These problems are solved by taking a very detailed look at the importance of boundary conditions. The last part of chapter 7 is again a look at the structure of the module.

As is already mentioned in chapter 5 it is very unstable to iterate to self-consistency directly with the output potentials or charge-densities of the different modules. In most modern methods there is a step in the self-consistency procedure that mixes either potentials or charge-densities. This is precisely what the module `MIX` does. In chapter 8 we describe the two different mixing methods currently implemented in the code. Also introduced in chapter 8 is the muffin-tin potential. This "simpler" potential enables us to perform some calculations much faster, although these calculations are less accurate.

The ability to perform conductance calculations means that we have to implement some new ways of finding states. Since we want to use the so-called Bloch states, we will have to implement a way of finding these into the code. Also the method for finding the band-structure and in particular the real bands at the Fermi energy has to be improved upon. This leads to the module called `GROW.COND`.

Up to this point in this thesis we have always assumed spin-degeneracy by putting a factor of two in certain places to take care of the fact that there is always a spin-up and spin-down electron in a state. In chapter 10 we examine ways to construct a version of the method and computer code that enables us to perform calculations for spin-polarised systems, i.e. systems where the spin-up and spin-down electrons behave differently. The relevant changes to the four modules are discussed in this chapter.

Part Three

The third and final part of this thesis describes some results obtained with the computer code. The first chapter of part three, chapter 11, concentrates on the two systems that were used as test systems during the development of the code. Nothing really new is found in these results, but we show that we have a very powerful tool for examining surfaces of solids.

Chapter 12 is more or less a feasibility study. In this chapter a bit of theory is derived, to enable the calculation of the conductance through a realistic domain wall. These realistic domain walls have a finite thickness that can be several hundred layers. We show that for the domain walls of experimental thickness the conductance is more or less equal to the bulk conductance. This is in agreement with the adiabatic picture that was proposed for these systems in the past.

The next chapter tries to say something new about the much discussed and studied system of the copper/cobalt multi-layers or in this case more specifically the single copper/cobalt interface. Since the embedding method enables us to study a single interface instead of a artificially periodic stack of these interfaces, we are indeed able to calculate something new.

The last chapter examines whether the method works for a system which is neither a bulk, a surface or an interface. This is the system of two surfaces almost touching. This leads to a tunneling barrier in between the two surfaces. In chapter 14 we show that indeed the code is able to cope with these systems, where the charge-density can become quite small in the vacuum layers between the surfaces. This is also one of the first studies of a tunneling system where the entire system is treated as one quantum system. The conductances calculated are in excellent agreement with known theories about tunneling currents.

Samenvatting

Een van de problemen die kunnen optreden tijdens de berekening van de elektronische structuur van vaste stoffen is de lokale symmetrie-breking. Denk bijvoorbeeld aan een oppervlak. Daar is de translatie symmetrie van het systeem gebroken in de richting loodrecht op het oppervlak. Dit proefschrift beschrijft een methode die het effect van deze lokale symmetrie-breking in rekening brengt door inbedding van het systeem met gebroken symmetrie in een systeem met een hogere symmetrie.

Het proces van het ontwikkelen van een nieuwe methode kan worden opgedeeld in drie stappen. Logischerwijs is de eerste stap de mathematische afleiding van de methode. De volgende stap is de implementatie van de mathematische constructie in een computer programma. Dat computer programma wordt dan gebruikt voor berekeningen.

Voordat we ons concentreren op de nieuwe methode beschouwen we eerst enkele problemen die inherent zijn aan dit soort berekeningen. Dit wordt behandeld in hoofdstuk 1. Hier wordt uitgelegd hoe de Schrödinger vergelijking van het kristal met ongeveer 10^{24} wisselwerkende deeltjes kan worden vereenvoudigd tot een één-elektron Schrödinger vergelijking, de Kohn-Sham vergelijking. Vervolgens wordt daar de in dit proefschrift geïntroduceerde methode om het symmetrie-breking probleem op te lossen vergeleken met de meest gangbare methode, die eveneens kort wordt beschreven in hoofdstuk 1.

Deel Een

In het eerste deel van dit proefschrift zullen we ons concentreren op de theoretische aspecten van de methode. Hoofdstuk 2 is gewijd aan de algemene 'embedding' methode. Dit is een beschrijving van de methode die niet gebaseerd is op een bepaalde ruimtelijke beschrijving. De eerste helft van dit hoofdstuk legt samen met appendix A het idee achter 'embedding' uit. Zoals de naam suggereert komt de methode erop neer dat het symmetrie-gebroken systeem wordt *ingebed* in een systeem met hogere symmetrie. De tweede helft van hoofdstuk 2 beschrijft de mathematische structuur en geeft het verband tussen de 'embedding' potentiaal en de Greense functie. De 'embedding' vergelijking die wordt geïntroduceerd in 2.3 vormt de basis van dit hele proefschrift.

Hoofdstuk 3 beschrijft de 'embedding' methode voor een verzameling van systemen met een bepaalde ruimtelijke structuur, namelijk een structuur die kan worden opgebouwd uit lagen. Dit is de beschrijving bij uitstek voor systemen als oppervlakken en grensvlakken. In dit hoofdstuk wordt een verbinding gelegd tussen 'embedding' theorie en verstrooiingstheorie. Oorspronkelijk is dat gedaan als een manier om de 'embedding' potentiaal efficiënter te berekenen, maar zoals blijkt in hoofdstuk 4, kunnen de verstrooiingseigenschappen worden gebruikt om heel interessante berekeningen te doen aan geleidbaarheidsproblemen.

In hoofdstuk 4 wordt de aanpak van geleidbaarheidsproblemen tot in details uitgewerkt.

Gebruikmakend van het Landauer formalisme wordt aangetoond dat de verstrooiingseigenschappen kunnen worden gebruikt om de geleidbaarheid uit te rekenen van een stapeling van lagen (bijv. een grensvlak). Zoals wordt aangetoond in dit hoofdstuk, leidt het bestuderen van elektron-transport tot allerlei interessante, nieuwe problemen betreffende de basis functies van het systeem.

Deel Twee

Het tweede deel concentreert zich op de implementatie van de methode in een computer programma. Gebruik makende van de programmeertaal FORTRAN (FORmula TRANslation) zou dit een vrij rechttoe rechtaan proces moeten zijn. Echter zoals hier beschreven blijkt dit niet het geval. In hoofdstuk 5 wordt de logische structuur van de methode beschreven en opgebroken in drie logische modulen: `GROW`, `FISH` en `MIX`.

De implementatie van de `GROW` module van de methode wordt beschouwd in hoofdstuk 6. Deze module lost de Schrödinger vergelijking op, met een potentiaal als invoer en uiteindelijk een ladingsdichtheid als uitvoer. In dit hoofdstuk wordt de keuze van een basis besproken, alsmede de afleiding van de matrix-elementen van de Hamiltoniaan (zie ook appendix B). De matrix van de Hamiltoniaan wordt gebruikt om de matrix van de Greense functie te construeren. Daarmee worden interessante eigenschappen zoals bijvoorbeeld de toestandsdichtheid berekend. In plaats van het beschrijven en opnoemen van alle details van de implementatie, worden enkel bepaalde belangrijke details onderzocht. De hoofdlijnen van de structuur van de `GROW` module worden aan het eind van het hoofdstuk gegeven.

In hoofdstuk 7 wordt de `FISH` module van het programma behandeld. Deze module heeft een ladingsdichtheid als invoer en gebruikt deze om de Poisson vergelijking op te lossen en zo een potentiaal te genereren als uitvoer. Een aantal problemen doen zich voor, die een direct gevolg zijn van de keuze voor systemen met een laag-structuur. Deze problemen worden opgelost door in detail te kijken naar het belang van randvoorwaarden. Het laatste gedeelte van hoofdstuk 7 geeft nogmaals de hoofdlijnen van de structuur van de module weer.

Zoals wordt vermeld in hoofdstuk 5 is het verkrijgen van een zelf-consistente oplossing via iteratie door direct gebruik te maken van de uitgevoerde potentialen en ladingsdichtheden van de verschillende modulen zeer instabiel. In de meeste moderne methodes is er daarom in dit iteratieproces een stap ingelast waar potentialen of ladingsdichtheden worden gemengd. Dit is de taak van de module `MIX`. Het programma bevat twee verschillende manieren om te mengen. Deze worden besproken in hoofdstuk 8. Ook wordt hier de 'muffin-tin' potentiaal geïntroduceerd. Deze 'simpelere' potentiaal zorgt ervoor dat bepaalde berekeningen een stuk sneller gedaan kunnen worden, hoewel deze berekeningen minder nauwkeurig zijn.

Om geleidbaarheidsberekeningen te kunnen doen, moet een nieuwe methode worden geïmplementeerd om toestanden te vinden. Het ligt voor de hand om Bloch toestanden te gebruiken. Derhalve moet een procedure worden geïntroduceerd in het programma om deze toestanden te berekenen. Met name aan het Fermi oppervlak. Al deze veranderingen leiden tot de nieuwe module `GROW.COND`.

In hoofdstuk 1 tot en met 9 wordt spin beschouwd als gedegenereerd door een factor twee toe te voegen waar nodig, om rekening te houden met het feit dat zich in elke toestand een 'spin-up' en een 'spin-down' elektron bevinden. In hoofdstuk 10 worden manieren onderzocht om een versie van het programma te construeren waardoor het mogelijk wordt om berekeningen uit te voeren voor spin-gepolariseerde systemen, d.w.z. systemen waar de 'spin-up' en 'spin-down' elektronen zich verschillend gedragen. De relevante veranderingen aan de vier

modules worden beschreven aan het einde van hoofdstuk 10.

Deel Drie

Het derde en laatste deel van dit proefschrift beschrijft resultaten van berekeningen met het programma. In het eerste hoofdstuk van dit deel, hoofdstuk 11, concentreren we ons op de twee test-systemen die tijdens de ontwikkeling van het programma zijn gebruikt. De resultaten zijn niet nieuw, maar tonen duidelijk aan dat we een zeer krachtig werktuig bezitten om oppervlakken en vaste stoffen te onderzoeken.

Hoofdstuk 12 is min of meer een haalbaarheidsstudie. Hier wordt een nieuw stukje theorie ontwikkeld, waarmee de geleidbaarheid kan worden berekend voor realistische domein-wanden. Deze domein-wanden hebben een eindige dikte van soms wel enkele honderden lagen. Aangetoond wordt dat voor domein-wanden met de experimentele dikte een geleidbaarheid wordt gevonden die min of meer gelijk is aan die van een stuk materiaal zonder domein-wand. Dit is in overeenstemming met het adiabatisch beeld dat in het verleden is voorgesteld voor dit soort systemen.

Het volgende hoofdstuk behandeld het veelbesproken en bestudeerde systeem van de koper/kobalt multilagen en geeft nieuwe resultaten voor het geval van één enkel grensvlak tussen koper en kobalt. Dit nieuwe resultaat is te danken aan de 'embedding' methode, die het mogelijk maakt om één enkel grensvlak te bereken.

Tot slot wordt in het laatste hoofdstuk onderzocht of de methode ook werkt voor een systeem dat niet valt binnen de eerder vermelde categoriën. Dit nieuwe systeem bestaat uit twee, elkaar bijna rakende oppervlakken, hetgeen leidt tot een zogenaamde tunnel-barrière tussen de beide oppervlakken. Het blijkt dat het programma inderdaad ook dit soort systemen aankan, ondanks het feit dat de ladingsdichtheid in het vacuüm tussen de oppervlakken zeer klein kan worden. N.B. dit is één van de eerste studies van een tunnel-systeem waar het hele systeem beschouwd wordt als één enkel quantum-systeem. De geleidbaarheden die zijn uitgerekend komen uitstekend overeen met de bekende theoriceën over tunnel-stromen.

Bibliography

[1] P. Hohenberg and W. Kohn, *Phys. Rev. B* **136**, 864 (1964)

[2] N.W. Ashcroft and N.D. Mermin, *Solid State Physics*, HRW International Editions, Hong Kong, 1976

[3] J.E. Inglesfield and G.A. Benesh, *Phys. Rev. B* **37**, 6682 (1988)

[4] J.B.A.N. van Hoof, *The Surface Embedded Green Function Method*, doctoraalscriptie, Nijmegen, 1991

[5] S. Crampin, J.B.A.N. van Hoof, M. Nekovee, J.E. Inglesfield, *J. Phys.: Condens. Matter* **4**, 1475 (1992)

[6] S. Crampin, M. Nekovee, J.B.A.N. van Hoof and J.E. Inglesfield, *Surf. Sci.* **287/8**, 732 (1993)

[7] X.-G. Zhang, P.J Rous, J.M. MacLaren, A. Gonis, M.A. Van Hove and G.A. Somerjai , *Surf. Sci.* **239**, 103-118 (1990)

[8] R. Landauer, *IBM J.Res.Dev.* **1**, 223 (1957) ; *Phils.Mag.* **21**, 863 (1970)

[9] C.J.W. Beenakker and H. van Houten, *Solid State Phys.* **44**, 1 (1991)

[10] M. Weinert, *J. Math. Phys* **22**, 2433 (1981)

[11] O.K. Andersen, *Phys. Rev. B* **12**, 3060 (1975)

[12] R.Stadler, W. Wolf, R. Podloucky, G. Kresse, J. Furthmüller and J. Hafner, *Phys. Rev. B* **54**, 1729 (1996)

A. Biedermann, O. Genser, W. Hebenstreit, M. Schmid, J. Redinger, R. Podloucky and P. Varga, *Phys. Rev. Letts.* **76**, 4179 (1996)

S. Blügel, *Phys. Rev. B* **51**, 2025 (1995)

E. Vescovo, O. Rader, J. Redinger, S. Blügel and C. Carbone, *Phys. Rev. B* **51**, 12418 (1995)

[13] C. Cohen-Tannoudji, B. Diu, F. Laloë, *Quantum mechanics*, Wiley Interscience, 1977

[14] A. Messiah, *Quantum mechanics*, North Holland, 1961

[15] J.B.A.N. van Hoof, S. Crampin and J.E. Inglesfield, *J. Phys.: Condens. Matter*, **4**, 8477-8488 (1992)

[16] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, *Numerical Recipes: The art of scientific computing*, section 9.3, Cambridge University Press

[17] Richard P. Brent, *Algorithms for Minimization without Derivatives*, Chapters 3 and 4, 1973

[18] E. Madelung, *Z. Phys.* **19**, 524 (1918)

[19] P.P. Ewald, *Ann. Phys.* **64**, 253 (1921)

[20] J.D. Jackson, *Classical Electrodynamics*, 2nd edition, Wiley, New York, 1975

[21] W.E. Baylis and S.J. Peel, *Comp. Phys. Comm.* **25**, 7 (1982)

[22] W. Kohn and L.J. Sham, *Phys. Rev.* **140**, A1133 (1965)

[23] D. Pines, *Elementary Excitations in Solids*, Benjamin Press, New York, 1963

[24] L. Hedin and B.I. Lundqvist, *J. Phys. C* **4**, 2064 (1971)

[25] U. von Barth and L. Hedin, *J. Phys. C* **5**, 1629 (1972)

[26] D.M. Ceperly and B.J. Alder, *Phys. Rev. Lett.* **45**, 566 (1980)

[27] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, *Numerical Recipes: The art of scientific computing*, section 2.6, Cambridge University Press

C.L. Lawson and R. Hanson, *Solving Least Squares Problems*, Chapter 18, Prentice Hall, New Jersey, 1974

[28] C.G. Broyden, *Math. Comput.* **19**, 577 (1965)

[29] G.P. Srivastava, *J. Phys. A* **17**, L317 (1984)

[30] H. van Leuken, *Electronic structure of metallic multilayers*, PhD-thesis, Amsterdam, 1991

[31] Stephen Gasiorowicz, *Quantum mechanics*, Wiley International, 1974

[32] H.B. Michaelson, *J. Appl. Phys.* **48**, 4729 (1977)

[33] J.E. Inglesfield, *Surf. Sci.* **76**, 355 (1978)

[34] Band-structures calculated using the old embedding code and using a layer Korringa Kohn Rostoker code provided by S. Crampin.

[35] Chikazumi, *Physics of Magnetism*, John Wiley & Sons, Inc., New York, 1964

[36] D. Jiles, *Introduction to Magnetism and Magnetic Materials*, Chapman and Hall, London, 1991

[37] C.W. Chen, *Magnetism and Metallurgy of Soft Magnetic Materials*, North Holland, page 84, 1977

[38] L.M. Sandratskii and J. Kübler, *Phys. Rev. B* **47**, 5854-5860 (1993)

[39] M. Uhl, L.M. Sandratskii and J. Kübler, *J. Magn. Magn. Mater.* **103**, 314-324 (1992)

[40] J. Kübler, K.-H. Höck, J. Sticht and A.R. Williams, *J. Appl. Phys.* **63**, 3482-3486 (1988)

[41] L. Berger, *J. Appl. Phys* **49**, 2156-2161 (1978)

[42] G.R. Taylor, A. Isin and R.V. Coleman, *Phys. Rev.*, **165**, 621 (1968)

[43] R.V. Coleman, R.C. Morris and D.J. Sellmyer, *Phys. Rev. B*, **8**, 317 (1973)

[44] K.M. Schep, P.J. Kelly and G.E.W. Bauer, *Phys. Rev. Lett.*, **74**, 586-589 (1995)

[45] Charles Kittel, *Introduction to Solid State Physics*, Wiley, (1986)

[46] K.M. Schep, J.B.A.N. van Hoof, P.J. Kelly, G.E.W. Bauer and J.E. Inglesfield, *to be published*

[47] K.M. Schep, P.J. Kelly and G.E.W. Bauer, *Mat.Res.Soc.Symp.Proc.* **384**, 305 (1995)

[48] E.L. Wolf, *Principles of Electron Tunneling Spectroscopy*, Oxford University Press, 1985

[49] Edwin A. Abbott, *Flatland*, Penguin, 1986

[50] D.D. Koelling and B.N. Harmon, *J. Phys. C* **10**, 3107 (1977)

[51] J.D. Bjorken and S.D. Drell, *Relativistic Quantum Mechanics*, page 55, McGraw-Hill, 1964

Curriculum vitae

Jeroen Bastiaan Antonius Nicolaas van Hoof
geboren op 30 juli 1969 te Oudenbosch

1981-1987	Thomas More College te Oudenbosch
1987-1991	Studie Natuurkunde aan de Katholieke Universiteit Nijmegen
1990/1991	Afstudeerwerk binnen de groep Electronic Structure of Materials onder begeleiding van Prof. dr. John Inglesfield. Titel afstudeerscriptie: *"The Surface Embedded Green Function method"*.
1991-1996	In dienst van de Stichting FOM als Onderzoeker in Opleiding. Promotie-werk aan de Katholieke Universiteit Nijmegen, in de Vakgroep Theoretische Fysica onder begeleiding van Prof. dr. John Inglesfield.
december 1996	In dienst van de Stichting FOM als Post-Doc.

List of publications

1. S. Crampin, J.B.A.N. van Hoof, M. Nekovee and J.E. Inglesfield, *Full-potential embedding for surfaces and interfaces*, J. Phys.: Condens. Matter **4**, 1475 (1992).

2. J.B.A.N. van Hoof, S. Crampin and J.E. Inglesfield, *The surface state–surface resonance transition on Ta(011)*, J. Phys.: Condens. Matter **4**, 8477 (1992).

3. S. Crampin, M. Nekovee, J.B.A.N. van Hoof and J.E. Inglesfield, *Subvolume embedding for interfacial electronic structure*, Surf. Sci. **287/8**, 732 (1993)

4. M.I. Trioni, J.B.A.N. van Hoof, S. Crampin, G.P. Brivio and J.E. Inglesfield, *The embedding method for surface adsorption*, Surf. Sci. **41** 307-309 (1994)

5. J.B.A.N. van Hoof, J.E. Inglesfield and I. Collins, *A new study of the surface electronic structure of W(001)* , Surface Review and Letters **1**, 685 (1994)

6. K.M. Schep, J.B.A.N. van Hoof, P.J. Kelly, G.E.W. Bauer and J.E. Inglesfield, *Interface Resistances in magnetic multilayers*, submitted

Printed in The Netherlands by Universal Press, Veenendaal